Naval Surface Warfare Center
Carderock Division
West Bethesda, MD 20817-5700

NSWCCD-65-TR–2003/02 November 2003

Survivability, Structures, and Materials Directorate

Technical Report

High-Volume Data Analysis Suite (HVDAS) Manual – November-17-2003

by

Gregory Alan Hildstrom - *The George Washington University*

20040422 132

DEPARTMENT OF THE NAVY

NAVAL SURFACE WARFARE CENTER, CARDEROCK DIVISION
9500 MACARTHUR BOULEVARD
WEST BETHESDA MD 20817-5700

9110
Ser 65-76
9 Mar 04

From: Commander, Naval Surface Warfare Center, Carderock Division
To: Distribution

Subj: HIGH-VOLUME DATA ANALYSIS SUITE (HVDAS) DOCUMENTATION

Ref: (a) RV *Triton* Seakeeping and Structural Full-Scale Trials: MOU contract number N0002499MD11140

(b) HSV-X1 Seakeeping and Structural Full-Scale Trials: Program Element 0603758N

(c) High Speed Sea Lift (HSS) Hydrodynamic Loads Model Testing: Program Element 0603564N

(d) DD-21 Hull Form Hydrodynamic Loads Model Testing: Program Elements 0603564N and 0604300N

Encl: (1) NSWCCD-65-TR-2003/02, *High-Volume Data Analysis Suite (HVDAS) Manual – November-17-2003*

1. References (a), (b), (c), and (d) requested the Naval Surface Warfare Center, Carderock Division (NSWCCD) to reduce and analyze full-scale sea trials or model test data. Enclosure (1) describes a new analysis tool which has been developed over the past two and a half years for this purpose. High-Volume Data Analysis Suite (HVDAS) is a collection of computer programs which take advantage of Linux clusters to increase the speed and efficiency of data analyses. HVDAS also uses batch processing to reduce the number of human errors in the data analysis process. The HVDAS documentation details the design, implementation, performance, and use of HVDAS for UNIX platforms.

2. Comments or questions may be referred to Mr. William Hay, Code 653; telephone (301) 227-1821; e-mail, HayWH@nswccd.navy.mil.

E.A. RASMUSSEN
By direction

Subj: HIGH-VOLUME DATA ANALYSIS SUITE (HVDAS) DOCUMENTATION

Distribution:

COMNAVSEASYSCOM WASHINGTON DC
 [SEA 05H (Waters), SEA 05P (Nappi, Packard)]

CNR ARLINGTON VA [ONR 334 Barsoum, Potter]

DTIC FORT BELVOIR VA

NAVSURFWARCEN CARDEROCKDIV
 BETHESDA MD [Codes 2420 (Kennel),
 2820 (Offutt), 3442 (TIC), 65, 651, 651 (Grassman),
 652, 653 (10), 653 (Brady) 654, 654 (Lewis), 655]

The George Washington University
 [Gregory A. Hildstrom]

Naval Surface Warfare Center
Carderock Division
West Bethesda, MD 20817-5700

NSWCCD-65-TR–2003/02 November 2003

Survivability, Structures, and Materials Directorate

Technical Report

High-Volume Data Analysis Suite (HVDAS) Manual – November-17-2003

by

Gregory Alan Hildstrom - *The George Washington University*

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE		3. DATES COVERED (From - To)
17-Nov-2003	Final		

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
High-Volume Data Analysis Suite (HVDAS) Manual - November-17-2003	N0002499MD11140
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER 0603785N and others

6. AUTHOR(S)	5d. PROJECT NUMBER
Gregory Alan Hildstrom	**5e. TASK NUMBER**
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Naval Surface Warfare Center Carderock Division 9500 Macarthur Boulevard West Bethesda, MD 20817-5700	NSWCCD-65-TR-2003/02

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
Naval Surface Warfare Center Carderock Division Code 65 9500 Macarthur Boulevard West Bethesda, MD 20817	**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This manual describes the design, implementation and use of the High-Volume Data Analysis Suite (HVDAS), which is a collection of parallelized statistical analysis software packages for Linux clusters. HVDAS is designed to reduce and perform statistical analysis on any raw time history data. This manual has been written for the HVDSAS version, November-17-2003, which was developed by the Naval Surface Warfare Center, Carderock Division, Code 653 and The George Washington University. This software is designed and built on a seven-node i686-based Linux cluster.

15. SUBJECT TERMS

ship structures, data reduction, data analysis, statistics

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	SAR	96	Mr. William H. Hay
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED			19b. TELEPHONE NUMBER (include area code) (301)-227-1821

Contents

List of Figures

List of Tables

1 Administrative Information

Funding for this project has come from a variety of model test and full scale sea trials programs. The High Volume Data Analysis Suite (HVDAS) has been used to speed the data analysis of the DD-21 and HSS model tests as well as the data analysis of the full scale sea trials of the Incat HSV-X1 and RV *Triton*.

HVDAS is not a specific deliverable; it is a tool developed from project funds to increase the speed and quality of data analysis.

1.1 DD-21 Hull Form Hydrodynamic Loads Model Testing

This analysis was performed by the Structures and Composites Department (Code 65) of the Survivability, Structures and Materials Directorate at the Naval Surface Warfare Center, Carderock Division (NSWCCD). The work was sponsored by Naval Sea Systems Command, SEA 05H and the DD-21 program office, PMS 500, under program element 0603564N, job order number 1-5500-114. Supplemental funding was provided by the Office of Naval Research under job order number 1-5500-159. This analysis was funded in FY00 under program element 0604300N, job order number 1-6530-456.

1.2 High Speed Sea Lift (HSS) Hydrodynamic Loads Model Testing

Hydrodynamic loads testing was completed on a segmented High Speed Sea Lift (HSS) Model 5594, scaled ratio 1:45, containing six shell sections connected using a calibrated backspline with instrumentation to measure primary and secondary hull girder loading. Similar to a trimaran, the extremely long slender center hull is stabilized by relatively short side hulls attached at an extreme aft location aligning all three hulls at the transom. This is part of a Navy program evaluating the structural design of a lightweight high-speed hull form as part of the High Speed Sea Lift program; program element 0603564N. Irregular wave experiments have been completed to evaluate primary and secondary structural loads and seakeeping performance. Model testing was performed at the NSWC Harold Saunders Maneuvering and Seakeeping (MASK) facility in June of 2002 and ended with irregular wave testing in July 2002 on carriage two of the David Taylor Model Basin.

1.3 HSV-X1 Seakeeping and Structural Full Scale Trials

The USNS Joint Venture HSV-X1 is an aluminum catamaran built by Incat. The HSV is produced commercially as a high speed ferry. Sea trials were carried out in 2001 and 2002 and involved trans Atlantic crossings, loading and unloading operations, helicopter landing and take off, and high speed combat support simulations. Funding was provided by two separate divisions of the Office of Naval Research. FY01 - 02: Expeditionary Logistics Future Naval Capabilities (FNC) High Speed Shuttle Product Line, program element 060XXXXN, job order number 01-2820-040. FY02 - 03: Swampworks, program element 0603758N, job order number 02-5500-195.

1.4 RV *Triton* Seakeeping and Structural Full Scale Trials

The RV *Triton* is a steel trimaran demonstrator research vessel built by Vosper Thornycroft in the UK. The sea trials were conducted in 2001 and 2002 and involved trans Atlantic crossings, speed and powering maneuvers, seakeeping sea trials, and structural sea trials. Funding was provided by a memorandum of understanding (MOU) contract between the US and UK. The MOU contract number is N0002499MD11140. The data reduction and analysis began in June of 2002 and continues through September of 2003. As of 2003, this is the largest full scale sea trials data analysis project ever attempted. The sea trials team recorded 50 star patterns, which consist of 10 different headings, in varying sea conditions. The largest peak wave amplitude experienced during the trials was 9.7 meters or 32 feet.

2 Acknowledgments

I have received a great deal of help understanding the statistical methods upon which HVDAS is being built. I wish to thank the following individuals who have provided valuable insights and assistance to my understanding of general test data analysis procedures, associated data acquisition hardware and software, and the development of this code and documentation.

- Thomas Brady, Code 653, NSWCCD, data analysis;

- Robert H Colegrove, Code 65, NSWCCD, editing;

- Daniel Gaies, Code 653, NSWCCD, data analysis, testing, HVDAS Data Extractor;

- J. Matthew Grassman, Code 651, NSWCCD, editing;

- William Hay, Code 653, NSWCCD, data analysis, editing;

- Paul Howden, Code 653, NSWCCD, editing;

- Richard Lewis, Code 653, NSWCCD, data analysis;

- Michael McDonald, Code 653, NSWCCD, coding, editing, algorithms, and testing; and

- Jesus Rosario, Code 653, NSWCCD, data acquisition.

3 Introduction

3.1 How To Use This Document

- Read this document in its entirety before performing any of the procedures.

- Check, Check, and Re-Check modifications or deletions of data and configuration files.

- Back up your data before doing anything else mentioned in this document.

- Back up your results before modifying or deleting data or configuration files.

3.2 Assumptions Made In This Document

The user has knowledge of and experience in:

- Time domain statistical analysis

- Frequency domain statistical analysis

- Linear transfer functions

- Discrete amplitude statistical analysis

- Weibull distribution statistics

- Binary and ASCII multiple-channel data files

- Format and location of the data files to be analyzed

- Finite impulse response filtering

- Linux operating system and a Unix shell

The user's analysis computer and user account have:

- The Local Area Multicomputer (LAM) implementation of Message Passing Interface (MPI) installed

- The core or core plus version of the Vector Signal Image Processing Library (VSIPL) installed

- Appropriate permissions to run MPI programs

- Appropriate permissions to access the data files to be analyzed

3.3 What Does HVDAS Do?

HVDAS is a collection of data analysis software algorithms designed to run in parallel on Linux clusters. HVDAS consists of the hvdalibrary and hvdas executable. The hvdas executable has command line flags for MMM, PSD, RAO, and Weibull analysis. Each of these analysis algorithms is designed to run as an automated process to analyze data in bulk. HVDAS requires the LAM-MPI library and VSIPL to compile from source code. Information about obtaining and installing these software packages can be found at http://www.lam-mpi.org and http://www.vsipl.org.

HVDALibrary contains most of the code that is common to the different analysis algorithms. HVDALibrary also contains most of the complex reusable analysis code used by each of the different analysis algorithms.

"hvdas -mmm" computes basic time domain statistics including mean, max, min, max-mean, min-mean, absolute range, maximum peak-to-peak amplitude, standard deviation, and kurtosis. MMM analysis can also convert data file formats and compute virtual channel data files.

"hvdas -psd" computes the Power Spectral Density[1, 2] of selected data channels, using the Finite Fourier Transform Method Segment Averaging Method (PSDFFTMSAM[1]).

"hvdas -rao" computes the Square Root Response Amplitude Operator, which is computed by $\sqrt{RAO} = \sqrt{\frac{PSDFFTMSAM(C_1)}{PSDFFTMSAM(C_2)}}$, which is simply the frequency based square root of the ratio of two power spectral densities. When analyzing surface ship structures, the RAO can calculate and provide ship response per unit of input wave height.

"hvdas -weibull" computationally estimates Weibull[16] distribution statistics on selected data channel amplitudes. You can choose the method of Weibull computation of either 2-Parameter Linear Regression, 2-Parameter Moment Method, 3-Parameter Linear Regression, or 3-Parameter Moment Method. These statistics are useful for calculating maximum lifetime loads.

"hvdas -convertdata" converts, scales, filters, or computes virtual channels and saves the output as tab-delimited ASCII text.

HVDAS provides the ability to apply scale factors, subtract initial condition

zero values, smooth, Finite Impulse Response[7] (FIR) filter, select data files, select channels, name channels, name conditions, select output locations, and read multiple data file formats. The algorithms in HVDAS are not in any way restricted to ship structure data.

4 What Does "hvdas -mmm" Do?

"hvdas -mmm" computes Mean Maximum Minimum (MMM) time domain statistics and is designed to reduce raw test data into a manageable spreadsheet of time domain statistical information. The user can configure the program to handle numerous filter types, data types, file formats, and test configurations. This software will compute the Mean, Max, Min, $(Maximum - Mean)$, $(Minimum - Mean)$, Range, Maximum Peak-to-Peak, Standard Deviation, Kurtosis, Mean Crossings, and $\frac{(Maximum-Mean)}{(StandardDeviation)}$. These statistics are calculated for each of five possible filter states on selected channels, files, and conditions. The user has the option of subtracting an initial zero condition value from every channel of data before analysis.

The results are also summarized by filter state and condition. The summary file contains weighted averages of statistics that do not require a discrete maximum or minimum value.

4.1 Mean

The Sample Mean[2] is calculated by $\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$. The Sample Mean $\bar{x}$ is an unbiased estimator of population mean.

4.2 Max

The Max is the maximum value that occurs in the time history.

4.3 Min

The Min is the minimum value that occurs in the time history.

4.4 Maximum-Mean

The Maximum-Mean is computed by subtracting the mean of the time history from the maximum value in the time history.

4.5 Minimum-Mean

The Minimum-Mean is computed by subtracting the mean of the time history from the minimum value in the time history.

4.6 Range

Range refers to the absolute range of the time history data. If the maximum value occurs after the minimum value in the time history, the range is computed by subtracting the minimum value of the time history from the maximum value of the time history. If the minimum value occurs after the maximum value, the range is computed by subtracting the maximum value of the time history from the minimum value of the time history. This can be useful when analyzing non-sinusoidal data, where a positive range value could indicate generally increasing data over time, while a negative range could indicate generally decreasing data over time. For sinusoidal data, the sign of the range indicates whether the minimum or the maximum value occurred first in the time history, which may or may not be useful.

4.7 Maximum Peak-to-Peak

Maximum Peak-to-Peak is the amplitude of the largest half cycle oscillation in the time history about the mean of the time history.

4.8 Variance

The Sample Variance[2] is calculated by $\sigma^2 = Var(x_1 \ldots x_N) = \frac{1}{N-1} \sum_{j=1}^{N} (x_j - \bar{x})^2$. The summation is divided by $N - 1$ because that makes the sample variance an unbiased estimator for the population variance since the mean of the population is estimated from the sample itself.

RV Triton VSTR116
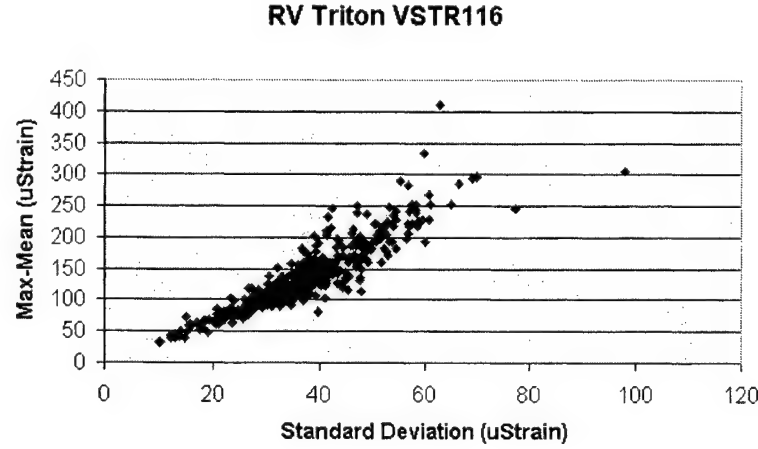
Figure 1: RV *Triton* MMM Standard Deviation Trend

4.9 Standard Deviation

The Standard Deviation[2] is calculated by $\sigma = \sqrt{\sigma^2}$. Taking the square root of the unbiased sample variance yields an unbiased estimator, the standard deviation. Figure 1 shows max-mean as a function of standard deviation for strain due to vertical bending on the RV *Triton*.

4.10 Kurtosis

The Kurtosis[4, 5] is calculated by $Kurt(x_1 \ldots x_n) = \frac{1}{N} \sum_{j=1}^{N} \left[\frac{x_j - \bar{x}}{\sigma} \right]^4$. This method is also known as the Pearson Kurtosis, which defines the Kurtosis of a normal distribution as 3. Division by N is required because all two parameter distributions are completely defined by the mean and the variance, which have already taken the degrees of freedom into account by dividing by N and N-1 respectively.

The Excess, defined as $Excess(x_1 \ldots x_n) = \frac{1}{N} \sum_{j=1}^{N} \left[\frac{x_j - \bar{x}}{\sigma} \right]^4 - 3$, is not the Kurtosis. The Excess, or Fisher Kurtosis, defines the Kurtosis of a normal distribution as 0, which is not the intended calculation and is not calculated by HVDAS.
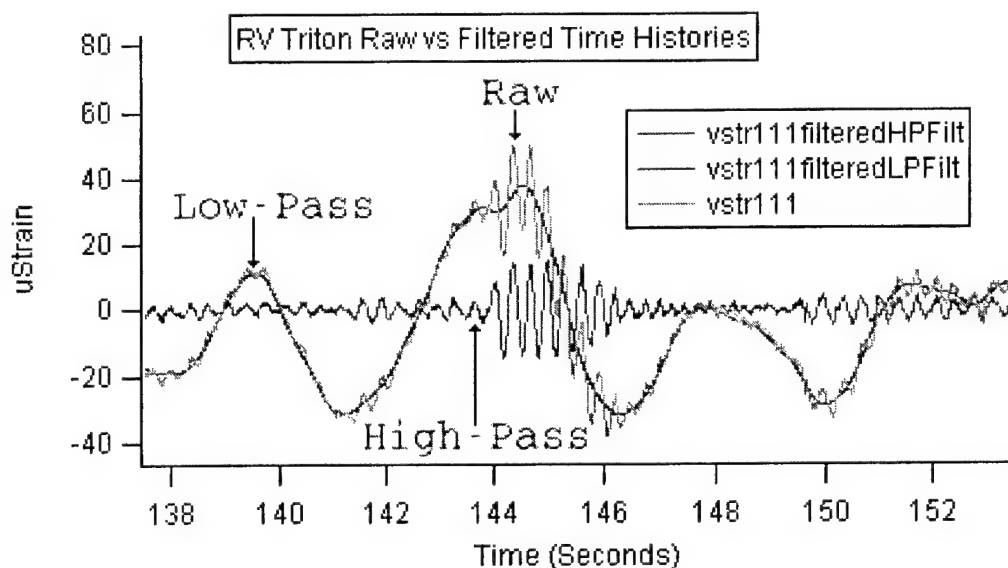
Figure 2: RV *Triton* Filtered Time History Comparison

4.11 Mean Crossings

Mean Crossings is the number of times the time history data oscillates about its mean.

4.12 (Maximum-Mean)/(Standard Deviation)

This value is computed by subtracting the mean from the maximum value in the time history and then dividing by the standard deviation of the time history. This value is a good check to quantify the severity of slamming, noise, or drop-outs in the data. This value defines how many standard deviations the maximum value is from the mean.

4.13 Filtering, Smoothing, and Zero Subtraction

Figure 2 shows raw, low-pass FIR, and high-pass FIR filtered versions of a time history, which was taken from RV *Triton* sea trials. The filters were designed using Wave Metrics IGOR Pro IFDL and the time histories were filtered with HVDAS.

18
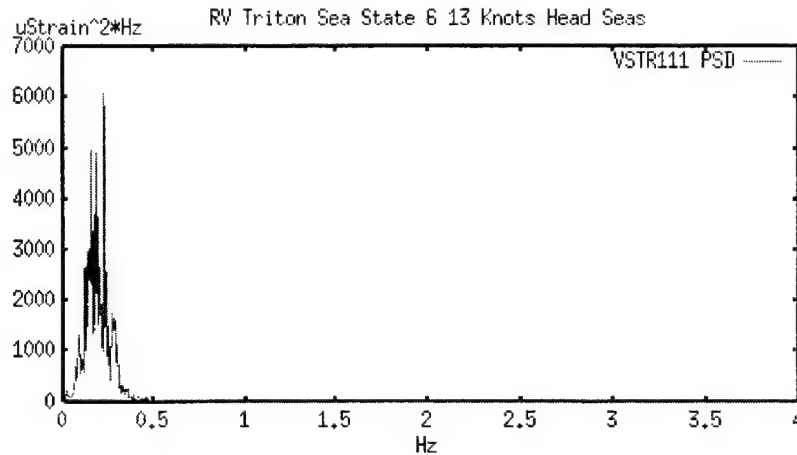
Figure 3: RV *Triton* PSD 0 to 4 Hz

5   What Does "hvdas -psd" Do?

"hvdas -psd" computes power spectral density (PSD) frequency domain statistics and is designed to reduce raw test data into a manageable spreadsheet of frequency domain information. The user can configure the program to handle numerous filter types, data types, file formats, and test configurations. These statistics are calculated for either raw or low-pass filter states on selected channels, files, and conditions.

The results are also summarized by filter state and condition. The summary file contains weighted averages of spectral ordinates.

Figure 3 shows an example PSD from the RV *Triton* sea trials data. The PSD has been plotted from 0 to 4 Hz, which encompasses wave-induced energy and first mode vibratory response, but the first mode energy is so small it is not visible on the current y-axis scale.

Figure 4 shows the same PSD plotted in from 0 to 0.6 Hz, which shows the wave-induced energy more clearly.

Figure 5 shows the same PSD plotted in from 2 to 4 Hz, which shows us the first mode vibratory response to the wave energy. The peak first mode frequency of 2.91 Hz is clearly visible; this is the natural frequency of the RV *Triton* in the vertical longitudinal plane.

19
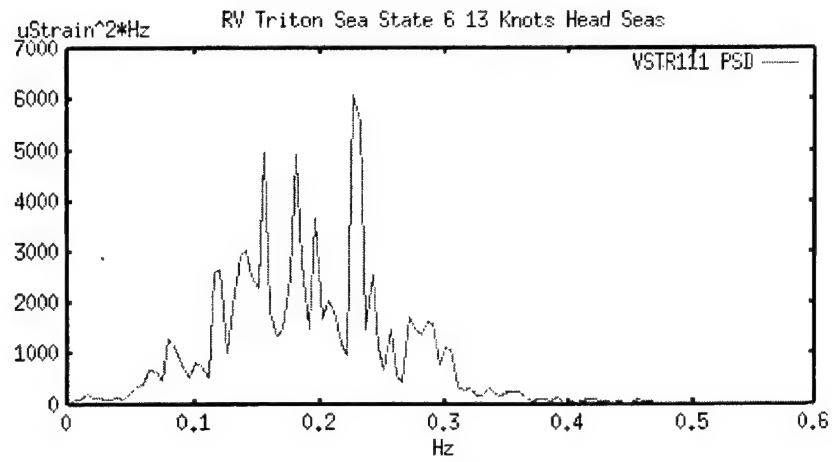
Figure 4: RV *Triton* PSD 0 to 0.6 Hz
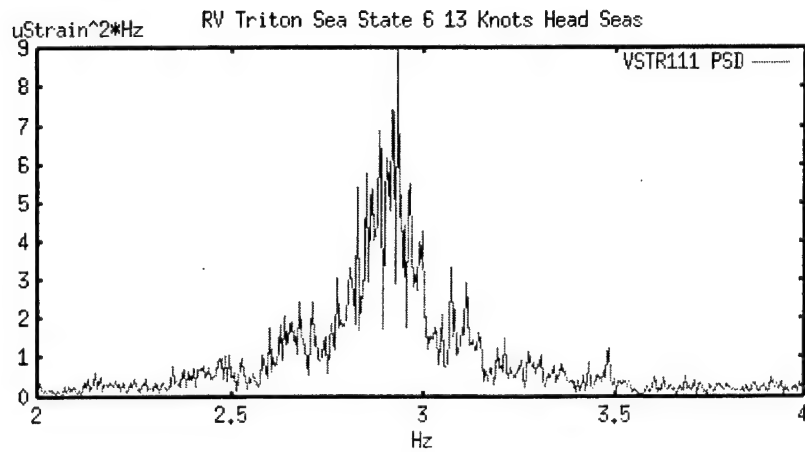
Figure 5: RV *Triton* PSD 2 to 4 Hz

5.1 Power Spectral Density Finite Fourier Transform Method Segment Averaging Method

This software will compute the power spectral density[1, 6, 2], which has user selectable resolution and frequency content.

The power spectra in this analysis use the Power Spectral Density Finite Fourier Transform Method Segment Averaging Method (PSDFFTMSAM[1]) with a 50% overlap.

The PSDFFTMSAM computes the Power Spectral Density using the Finite Fourier Transform Method (PSDFFTM) on a number of segments of the original data channel, which have the mean of each segment subtracted out. The local segment mean subtraction prevents extremely low frequencies from aliasing to dc in the output spectrum. The output PSD will be $\frac{SegmentLength}{2} + 1$ points long. Each segment PSD is summed and divided by the total number of segments in the input data channel, which includes the overlap. This summation and division results in an average PSD for the data run.

The computed PSDFFTM uses I-Skipping to achieve greater resolution. The amount of I-Skip and the PSD segment length are computed automatically based on the Original Sample Rate of the data, Highest Frequency of Interest (HFI) of the output PSD, amount of Oversampling of the HFI, and the Target Delta Frequency of the output PSD, which are all selectable by the user. The I-Skipped data vector is computed by averaging the point to be extracted with the points to be skipped, which eliminates high frequency aliasing.

If any single data channel contains too few data points to compute the PSD with the desired resolution and frequency content, that channel will be repeated and concatenated with itself until enough points exist to perform the PSD computations. This method assumes each short data run would continue with identical frequency content until enough data existed to compute the desired PSD. Any discontinuities exist as frequencies above the Nyquist frequency, which is half of the sample rate, and will not show up in the output PSD.

The Finite Fourier Transform Method (FFTM) of computing the PSD assumes that $h(t) = RealInputDataVector$ and that $H(f) = FourierTransform(h(t))$. The PSDFFTM is computed by $PSDFFTM(f) = 2*\mid H(f) \mid^2$. The FT is performed on the original data vector segment. We use the real and imaginary parts of the FT to create a real magnitude squared FT. We then multiply by two to account for the one sided spectrum, then divide by the input data vector size and the sample rate to scale the power spectrum. This last step needs to be done because of the FT implemen-

tation we use. The area under the final PSD is the variance of the original time history. The corresponding frequency of each PSD magnitude squared is calculated using the number of PSD points, the number of the current point, and the Nyquist frequency, which is half of the sample rate.

6 What Does "hvdas -rao" Do?

"hvdas -rao" computes Square Root Response Amplitude Operator (RAO) and is designed to reduce raw test data into a manageable spreadsheet of frequency domain information. The user can configure the program to handle numerous filter types, data types, file formats, and test configurations. RAOs are calculated for either raw or low-pass filter states on selected channels, files, and conditions.

The results are also summarized by filter state and condition. The summary file contains weighted averages of RAO ordinates.

Figure 6 shows an RAO from the RV *Triton* sea trials in the Sea State 4, 10 knots, head seas condition. You can see that, at 0.3 Hz, the strain, due to vertical longitudinal bending, of the RV *Triton* has the largest response to input wave height. The max valid RAO encounter frequency for the RV Triton is probably 0.7 to 0.8 Hz. The true maximum linear response to wave energy occurs close to 0.3 Hz encounter. After 0.8 Hz the response is nonlinear vibration and the RAO transfer function is not useful. Because of this behavior, it is necessary to also look at structural and wave height PSDs separately to determine valid RAO frequency ranges.

6.1 Response Amplitude Operation Computation

This software will compute the Square Root RAO using two PSDs, which have user-selectable resolution and frequency content.

The power spectra computed in this analysis utilize the Power Spectral Density Finite Fourier Transform Method Segment Averaging Method (PSDFFTMSAM) with a 50% overlap, as described in the PSD analysis section.

An RAO is simply the frequency by frequency ratio of a spectra to an excitation spectra, $\left(\sqrt{RAO}\right)^2 = \frac{PSDFFTMSAM(DataChannel)}{PSDFFTMSAM(DivisorChannel)}$, where the RAO units depend on the units of the input data channels. In ship structures, the $\left(\sqrt{RAO}\right)^2 = \frac{PSDFFTMSAM(ResponseChannel)}{PSDFFTMSAM(WaveHeight)}$, which yields results in terms of ship response per unit of wave height, but the output transfer function depends

RV Triton Example Root RAO
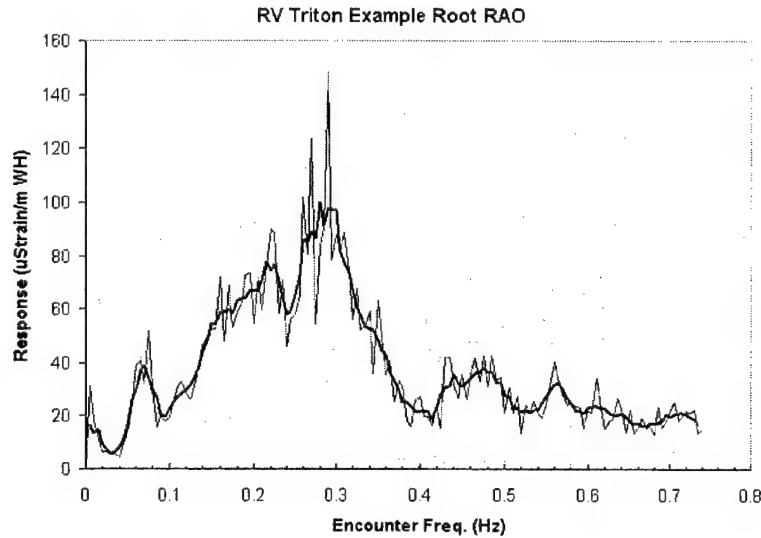
Figure 6: RV *Triton* RAO Example 0 to 0.8 Hz

on the units of your data channels. Users can use the optional conversion from encounter to wave frequency, which is based on the current condition's speed and heading, to convert RAOs from encounter (observed) frequency to wave frequency.

6.2 Encounter to Wave Frequency Conversion

Encounter frequency spectra show frequencies as they were observed or encountered on the ship. Converting to wave frequency adjusts spectra frequencies to give the same spectra frequencies as a stationary wave buoy, which essentially removes speed and heading effects of the ship movement.

The optional conversion from encounter to wave frequency[17, 18] is computed using $w_{wave} = \frac{-1+\sqrt{1+4\left(\frac{V\cos(\Theta)}{g}\right)w_e}}{2\left(\frac{V\cos(\Theta)}{g}\right)}$, where $g = 32.18$ ft/sec/sec, V = speed in ft/sec, $6076ft = 1 Nautical Mile$, Θ =heading in degrees, $HeadSeas = 0$ degrees, $w_e = f_e 2\pi$, and $f_w = \frac{w}{2\pi}$. Head seas means that the ship is heading straight into the waves. This equation works for all runs with headings from 0 to < 90 degrees and from > 270 degrees to 360 degrees. If the heading is beam seas, 90 or 270 degrees, the wave frequency is equal to the encounter frequency. The

23
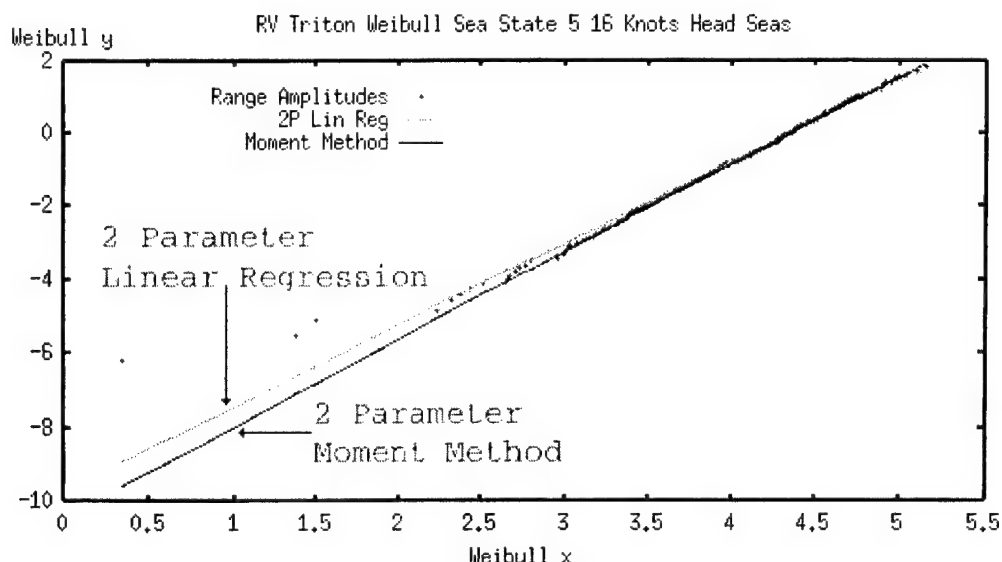
Figure 7: RV *Triton* Range Amplitudes Weibull

conversion in following and stern quartering seas, > 90 degrees to < 270 degrees, is not implemented because the solution can be discontinuous and indeterminate.

7 What Does "hvdas -weibull" Do?

"hvdas -weibull" computes Weibull[6] distribution statistics on a set of data amplitudes and is designed to reduce raw test data into a manageable spreadsheet of linearized Weibull distribution parameters. These statistics are calculated for either raw, low-pass, or high-pass filter states on selected channels, files, and conditions. The results are computed on the entire set of amplitudes that exist in a single test condition, which yields one output file per condition for each variant of analysis. Weibull analysis linearizes the data amplitudes[6], in Weibull space, and computes trend lines for lifetime estimation[6].

Figure 7 shows linearized Weibull amplitudes and two corresponding solution methods: 2 parameter linear regression and 2 parameter moment method.

7.1 Solution Methods

7.1.1 Weibull Linearizing

The selected data amplitudes can be linearized into a $y = m*x + b$ format[6]. The Weibull Y-axis data can be represented by $\ln\left(\ln\left(\frac{1}{1-P(x)}\right)\right) = \beta \ln(x) - \beta \ln(\eta)$, where $P(x) = 1 - e^{-\left(\frac{x}{\eta}\right)^{\beta}}$. A linearized Weibull plot can be achieved by plotting the Weibull Y-axis data against $\ln(x)$ on the X-axis. If data amplitudes are Weibull in nature, this will yield a straight line in slope-intercept form where β is the slope and $(-\beta \ln(\eta))$ is the Y-intercept.

7.1.2 2-Parameter Linear Regression Method

1. Find data amplitudes,

2. Rank amplitudes in ascending order,

3. Linearize the ordered amplitudes in Weibull space,

4. Perform a linear regression, and

5. Determine the correlation coefficient.

7.1.3 Moment Method Iterative Calculation

The moment method[6] attempts to determine the line that best represents the Weibull data based completely on the mean and variance of the amplitudes. The shape, or slope, of the Weibull distribution can be completely defined by the mean and variance of the amplitudes; as shown here $\frac{\sigma_x^2}{\mu_x^2} = \frac{\Gamma\left(1+\frac{2}{\beta}\right)}{\left(\Gamma\left(1+\frac{1}{\beta}\right)\right)^2} - 1$. Iterating through this equation will yield β, the moment method shape parameter, which can be used along with the characteristic value from the population to determine a moment method line, which should better represent the higher magnitude Weibull amplitudes. The characteristic value can be determined by $\mu_x = \eta\left(\Gamma\left(1 + \frac{1}{\beta}\right)\right)$. Provided that the data are known to be Weibull, the moment method is a bias-free method of estimating the Weibull parameters.

7.1.4 2-Parameter Moment Method

1. Find data amplitudes,

25

2. Calculate the mean of the amplitudes (μ_x),

3. Calculate the variance of the amplitudes (σ_x^2),

4. Iteratively solve for beta, the shape parameter, and

5. Calculate the moment method line

7.1.5   3-Parameter Moment Method Iterative Calculation

The moment method[6] attempts to determine the line that best represents the Weibull data based completely on the mean and variance of the amplitudes. The general three-parameter Weibull distribution is $P = 1 - e^{-(\frac{x-x_0}{\eta-x_0})^\beta}$, where β is the Weibull slope or shape parameter, x_0 is the threshold value of the data, η is the characteristic value of the distribution of data, P is the cumulative probability, and x is the Weibull distributed data.

The mean and variance of Weibull data can be expressed as: $\mu_x = (\eta_x - x_0)\Gamma(1 + \frac{1}{\beta}) + x_0$ and $\sigma_x^2 = (\eta_x - x_0)^2\left[\Gamma(1 + \frac{2}{\beta}) - (\Gamma(1 + \frac{1}{\beta}))^2\right]$. The characteristic value can be estimated by choosing the value in the distribution with a cumulative probability as close as possible to 0.632. A bounding range for beta can be found using the following equation: $\frac{\mu_x - x_0}{\eta_x - x_0} = \Gamma(1 + \frac{1}{\beta})$. The threshold can range from 0 to the smallest actual data amplitude, which can be used in the previous equation to bound beta.

The Weibull slope can be solved for iteratively by balancing the left and right sides of the following equation to the desired level of accuracy.
$$\frac{\sigma_x^2}{\left[\mu_x - \frac{\eta_x\Gamma(1+\frac{1}{\beta}) - \mu_x}{\Gamma(1+\frac{1}{\beta}) - 1}\right]^2} = \frac{\Gamma(1+\frac{2}{\beta}) - \left[\Gamma(1+\frac{1}{\beta})\right]^2}{\left[\Gamma(1+\frac{1}{\beta})\right]^2}$$
This leaves only x_0 to be solved for in the following equation:
$$x_0 = \mu_x - \frac{\sigma_x\Gamma(1+\frac{1}{\beta})}{\left\{\Gamma(1+\frac{2}{\beta}) - \left[\Gamma(1+\frac{1}{\beta})\right]^2\right\}^{\frac{1}{2}}}.$$

7.1.6   3-Parameter Moment Method

1. Find data amplitudes,

2. Calculate the mean of the amplitudes (μ_x),

3. Calculate the variance of the amplitudes (σ_x^2),

4. Estimate the characteristic value (η_r),

5. Find bounds for beta,

6. Iteratively solve for beta,

7. Solve for the Weibull threshold (x_0), and

8. Calculate the moment method line

7.2 Amplitude Selection

7.2.1 Maximum Amplitudes

This method of amplitude selection uses peaks that occur above the mean of the time history in the Weibull calculation.

7.2.2 Minimum Amplitudes

This method of amplitude selection uses troughs that occur below the mean of the time history in the Weibull calculation.

7.2.3 Maximum - Mean Amplitudes

This method of amplitude selection uses all peaks - mean of the time history in the Weibull calculation. These amplitudes are determined by subtracting the mean of the time history from the maximum amplitudes.

7.2.4 Minimum - Mean Amplitudes

This method of amplitude selection uses all troughs - mean of the time history in the Weibull calculation. These amplitudes are determined by subtracting the mean of the time history from the minimum amplitudes.

7.2.5 Peak-to-Peak Range Amplitudes

This method of amplitude selection uses all local half-cycle peak-to-peak ranges (maximums - minimums) about the mean of the time history in the Weibull calculation. These amplitudes are determined by looking for oscillations about the mean value of the time history. The peak-to-peak amplitudes have double the

number of amplitudes as maximums or minimums because every half-cycle amplitude is counted, which means all peak-to-trough amplitudes and trough-to-peak amplitudes. This ensures that the Weibull distribution data contains the largest half-cycle event in the time history. One must be careful when using these amplitudes however. The number of peak-to-peak amplitudes is twice the number of wave encounters.

7.2.6 Raw Amplitudes

This method of amplitude selection treats the entire input data channel as a set of amplitudes. This means that you can use another program to generate or adjust the amplitudes and then analyze them in HVDAS. This allows another program, such as IGOR or Excel, to perform the amplitude selection. These amplitudes can then be run through HVDAS to have Weibull analysis performed on them.

7.2.7 Whipping Amplitudes

This method of amplitude selection looks for football shaped series of consecutive high pass filtered amplitudes.

7.2.8 Impact Amplitudes

This method of amplitude selection looks for a certain duration event that is greater than a threshold value.

7.3 Filter Choices

This software has the ability to separate, for further Weibull analysis, frequency components of a time domain signal.

7.3.1 Low Pass

The low-pass Finite Impulse Response (FIR) filter is usually used to remove high-frequency energy from the time domain signal. Use it to yield the quasi-static ordinary wave-induced portion of the signal by removing higher frequency vibration, slamming, and whipping cycles. The amplitude selection steps are as follows.

1. Read in the raw data.

2. Low-pass FIR filter the raw data.

3. Find all mean crossings and relative maximums and minimums in the filtered time history.

4. Use the local maximums and minimums to generate local ranges, also called peak-to-peak values.

5. Select the proper amplitude set and perform the Weibull analysis.

7.3.2 High Pass

The high-pass FIR filter will remove low-frequency energy from the signal. Use it to yield the vibration, slamming, and whipping portion of the signal by removing low frequency ordinary wave-induced energy. The amplitude selection steps are as follows.

1. Read in the raw data.

2. Low-pass FIR filter the raw data.

3. Find all mean crossings in the low-pass FIR data to identify wave events experienced by the ship.

4. High-pass FIR filter the raw data.

5. Find relative maximums and minimums in the high-pass filtered time history using the mean crossing locations.

6. Use the local maximums and minimums to generate local ranges, also called peak-to-peak values.

7. Select the proper amplitude set and perform the Weibull analysis.

7.3.3 Raw

Raw is the absence of a filter and will not remove any energy from the signal. We usually use it to yield the wave+whipping portion of the signal. The amplitude selection steps are as follows.

1. Read in the raw data.

29

2. Low-pass FIR filter the raw data to identify wave encounters.

3. Find all mean crossings in the low-pass FIR data to identify wave events experienced by the ship.

4. Find relative maximums and minimums in the raw time history using the mean crossing locations.

5. Use the local maximums and minimums to generate local ranges, also called peak-to-peak values.

6. Select the proper amplitude set and perform the Weibull analysis.

7.3.4 Already Filtered

The already filtered option will a simple result in the Weibull calculation. The data is assumed to be already filtered. This removes the digital FIR filtering step from the amplitude selection process.

1. Read in the raw already filtered data.

2. Find all mean crossings and relative maximums and minimums in the already filtered time history.

3. Use the local maximums and minimums to generate local ranges, also called peak-to-peak values.

4. Select the proper amplitude set and perform the Weibull analysis.

7.4 Prediction

The main goal of Weibull analysis is to statistically characterize the data and to make predictions. Finite amounts of test data, that are Weibull in nature, can be analyzed with this software and yield response predictions based on estimates of many years or even a ship's entire lifetime.

There are two main types of predictions that we use Weibull analysis for: fatigue life and lifetime maximum value. For fatigue life prediction, a two or three parameter linear regression is usually sufficient, because it is the line that best matches the majority of amplitudes that occurred during the test. The two or three parameter Moment Method line usually does a better job of estimating maximum lifetime values.

For ship structures and wave data, wave induced behavior usually has a linear regression Weibull slope of around 2, corresponding to a Rayleigh distribution. Slamming and vibration behavior usually has a linear regression Weibull slope of around 1. Normal distribution Gaussian behavior, when treated as a Weibull distribution, usually has a linear regression Weibull slope of around 3.5.

7.4.1 Parent Weibull Distribution Definition

The Parent Weibull Distribution Definition is $P(x) = 1 - e^{-\left(\frac{x}{\eta}\right)^{\beta}}$. For real data, P is estimated by $\frac{m}{n+1}$, where n is the actual number of real data amplitudes from the trial or model test. For estimated lifetime data, P is estimated by $\frac{M}{N+1}$, where N is the estimated number of data points that would be generated in a lifetime exposure to a given single condition. P is the most probable value at N for an entire lifetime of exposure. The expected lifetime value $x = \eta \left[-ln\left(1 - \frac{M}{N+1}\right)\right]^{\frac{1}{\beta}}$.

7.4.2 Extreme Value Distribution of the Ship Lifetime Max

The Extreme Value Distribution of the ship lifetime max for the Nth point is $F(N) = \left[1 - e^{-\left(\frac{x}{\eta}\right)^{\beta}}\right]^{N}$. For multiple ship lifetimes, like a class of ships, P(N) would have a distribution of its own. To find the extreme value of P that is exceeded in only 1 of 10 lifetimes, we would set $F(N) = 0.9$ and solve for x. The extreme value $x = \eta \left[-ln\left(1 - F^{\frac{1}{N}}\right)\right]^{\frac{1}{\beta}}$.

7.5 Weibull Example Problem

Assume that we have determined the shape parameter $\beta=2$ and the characteristic value $\eta=10$ from a moment method or linear regression Weibull analysis of some time history data. Now assume that a lifetime exposure to the condition being analyzed would produce a population of 500 points. To find the estimate of the "most probable extreme value" of this 500th point, solve for x in the following equation. $ln\left(ln\left(\frac{1}{1-\frac{500}{501}}\right)\right) = 2ln(x) - 2ln(10)$, $1.8272 = 2ln(x) - 4.6052$, which yields $x = 24.93$. To find the value in the distribution associated with the 500th point that is exceeded only 1 in 10 times; $0.9^{\frac{1}{500}} = 1 - e^{-\left(\frac{x}{10}\right)^{2}}$, $e^{-\left(\frac{x}{10}\right)^{2}} = 0.000211$, which yields $x = 29.095$. However, if you plug 24.93 into the extreme value equation and solve for $F(N)$, you will find that the most probable value of 24.93 has a 63.23% chance of being exceeded. It is likely that the designer will

31
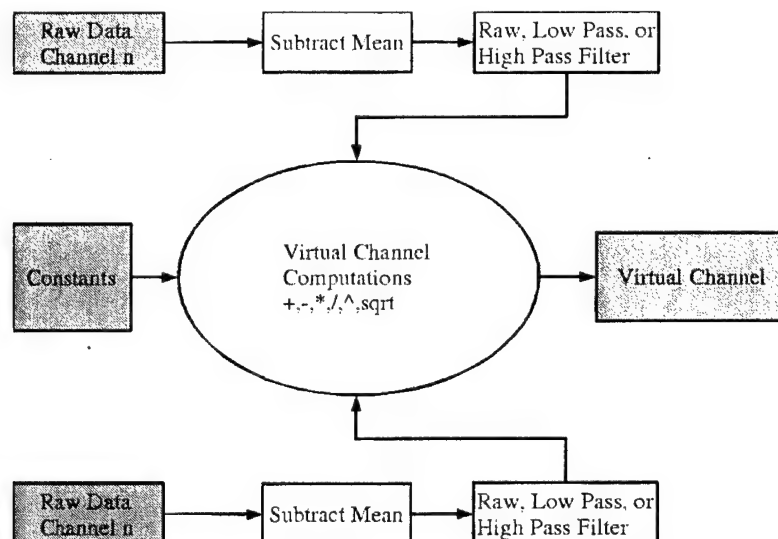
Figure 8: Virtual Channel Block Diagram

want a much smaller chance of the predicted extreme value being exceeded. For example, typical values selected for $F(N)$ might be 0.9, 0.95, 0.99, or 0.999. The choice of this value should be representative of the size of the population and the consequence of exceeding the maximum value.

8   What Does "hvdas -convertdata" Do?

"hvdas -convertdata" can use the MMM configuration settings to modify and convert the original input data. Data conversion can be used to filter, smooth, scale, subtract initial condition zero values, or compute virtual channels from the original input data. Additionally, this operation can be performed on binary or ASCII input data. The output will be in ASCII tab-delimited text format.

8.1   Virtual Channels

Figure 8 shows a block diagram describing the procedure used to create virtual channels.
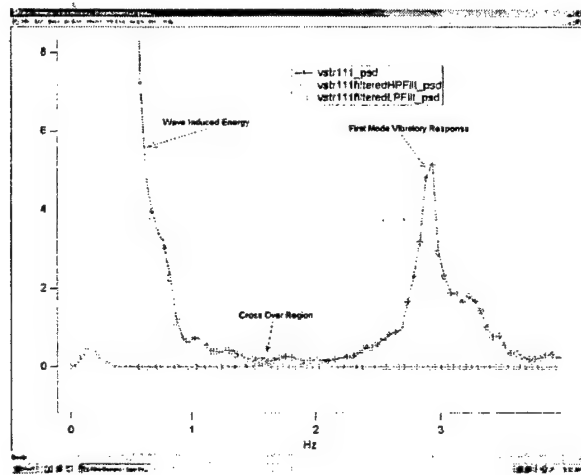
Figure 9: Crossover Region Filter Design Work

9 FIR Filter Design

HVDAS does not have any built-in limitations when it comes to Finite Impulse Response (FIR) filters. HVDAS imports its filters as lists of filter coefficients, which need to be generated by some other program, such as Wave Metrics IGOR Pro IFDL. A higher number of filter coefficients is not necessarily better. Filters with high numbers of coefficients generally have a larger impulse response, which means that the output exhibits a "ringing" filtering impulse-like events and consequentially produce an impulse response, which could be larger than the data being analyzed. The number of filter coefficients also places requirements on the size of data file that HVDAS can filter. HVDAS requires data files to have more scans than the number of filter coefficients. It is mathematically impossible to apply a 1500 coefficient FIR filter to a data file that contains only 500 or 1000 samples.

For most ship structures, there is a significant difference between the directly wave-induced frequency response and the first-mode vibratory frequency response. This frequency gap can be advantageous when designing FIR filters. The example, shown in Figure 9, came from RV *Triton*, which is a full scale trimaran demonstrator ship. The channel chosen is VSTR116, which is the midship keel gage measuring axial strain due to vertical longitudinal bending. Figure 9 shows PSDs of raw, low-pass FIR filtered, and high-pass FIR filtered data in Wave Metrics IGOR Pro. The crossover frequency of 1.55 Hz was chosen because it is about
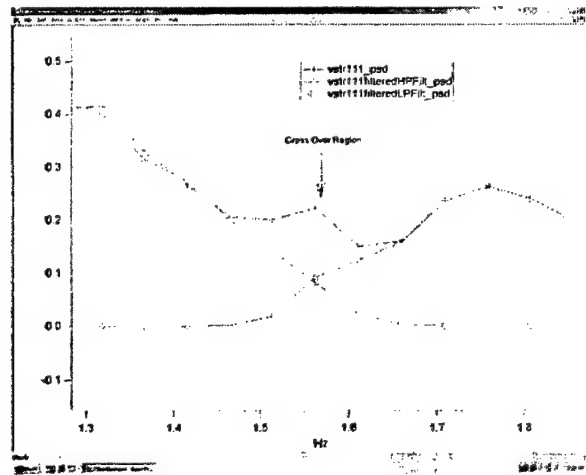
Figure 10: Crossover Region Close Up

half way between the wave-induced energy and the first mode vibratory response. The crossover frequency needs some frequency space on either side of it because FIR filters are not perfect and do not have immediate or infinite attenuation. Figure 10 shows a close up of the crossover region. Figure 11 shows the raw, low-pass FIR filtered, and high-pass FIR filtered time histories.

The crossover region can be a single frequency step or a wide range of frequencies. Different data may not even need the filters to cross over in the frequency domain.

10   Requirements and Limitations

- All channel numbers used by this software are numbered starting from zero for referencing purposes. The first data channel will be referred to as channel '0' and the last data channel will be the number of total channels minus one. For example: if your data files have 54 total channels, you will need to refer to each individual channel as 0 through 53.

- Each block of data (the folders/files to be run with the same configuration files) must have the same channel numbering scheme throughout. For example, if you have chosen to analyze channel '3', then channel '3' must refer to the same channel throughout the data block for consistent results.
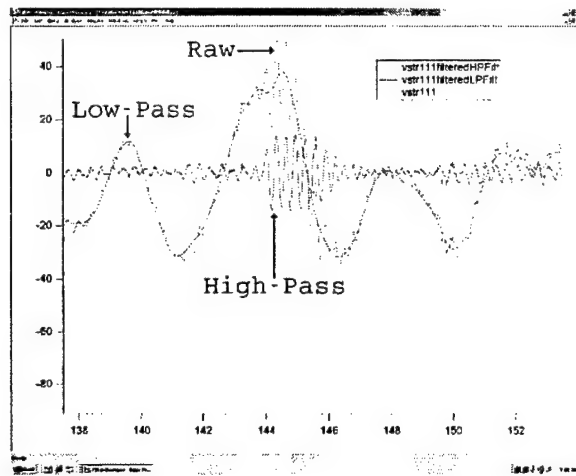
Figure 11: Raw, Low-Pass FIR, and High-Pass FIR Time Histories

- The only data types verified are binary, 4-byte, single-precision, high-byte-first interleaved data; 2-byte, unsigned-integer, low-byte-first interleaved data; and ASCII, tab-delimited, text data. 4-byte, single-precision, low-byte-first data files are also supported, but not heavily tested. Future versions will better accommodate multiple file formats.

- Data files in a single condition directory must have the same data format, number of channels, naming scheme and scale factors.

- This software assumes that your data files follow the numeric naming conventions: FILE-PREFIX-###-FILE-EXTENSION or FILE-PREFIX-#-FILE-EXTENSION.

- It is a good idea to stay away from spaces in file names. Not all systems handle spaces in file names the same way, which makes it difficult to code for compatibility across all systems. If you need to use a space in a file name or condition label, replace it with an underscore "_".

- This software assumes that input data files have at least 4-character file names.

- You must have LAM MPI installed to compile the source code.

- You must have VSIPL core or core plus to compile the source code.

Program/Library	Lines of Code	Executable/Compiled Size (Bytes)
hvdalibrary	18546	5447464
hvdas	13657	7398535

Table 1: Code and Executable Size Breakdown

- This software will not do everything for you. You must spend time getting familiar with your data and its layout. You will be required to make intelligent decisions about analysis options which can affect your results. A good way to do this is by using Wave Metrics IGOR Pro to look at and analyze small amounts of data. Once you know what you are doing, run a small batch of analysis, check the results to make sure you are getting what you want, and then run the entire batch of automated analysis.

- Time selection support has been tested with 1 or more files. Time selection start and end times must be input correctly. Times that exceed the end of your data files may not be properly detected and may cause the program to segfault.

- HVDAS will currently only scale up to the number of channels in your data files. If you try to analyze a set of data files that contain 100 channels each on a cluster with 128 processors, HVDAS will only use 101 processors; 100 client processors and 1 server processor.

11 Software Organization

11.1 General Software Overview

HVDAS is about 32,203 lines of original C++ code, including whitespace and comments, as of HVDAS version 2003-11-09. Printing HVDAS would require about 620 printed pages, assuming an average of 52 text lines per printed page on ISO A4 paper. This line count does not include any of the linked code in VSIPL or LAM-MPI, which were developed by other people. Figure 12 shows the relationship between the operating system, GNU GCC compilers, and HVDAS. Table 1 shows the breakdown of code and executable size among the different components of HVDAS.
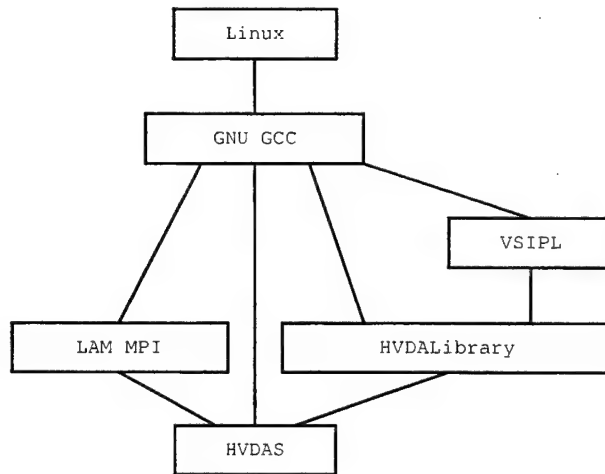
Figure 12: HVDAS General Software Overview

11.2 How Does HVDAS Analyze Data Files So Fast?

HVDAS uses a coarse parallelization to achieve its speed. At NSWCCD, Code 65, there are typically a large number of files and data channels to calculate statistics from. In the early design phases of HVDAS it was determined that speeding up bulk data analysis was more important than speeding up the analysis of a single channel. This led to the use of a single data channel, or time history, as the smallest piece of data to analyze. Breaking up the work by channel is a logical and highly parallel analysis method, but it does not help analyze a single data channel any faster than our uniprocessor analysis code.

Figure 13 shows the work flow distribution that allows HVDAS to perform and scale so well.

11.3 How Do The HVDAS Programs Analyze Data Files In Parallel?

11.3.1 Server Algorithm Overview

Figure 14 shows how the server portion of HVDAS loops, breaks up the calculations, distributes work to the client nodes, and receives results from the client nodes.
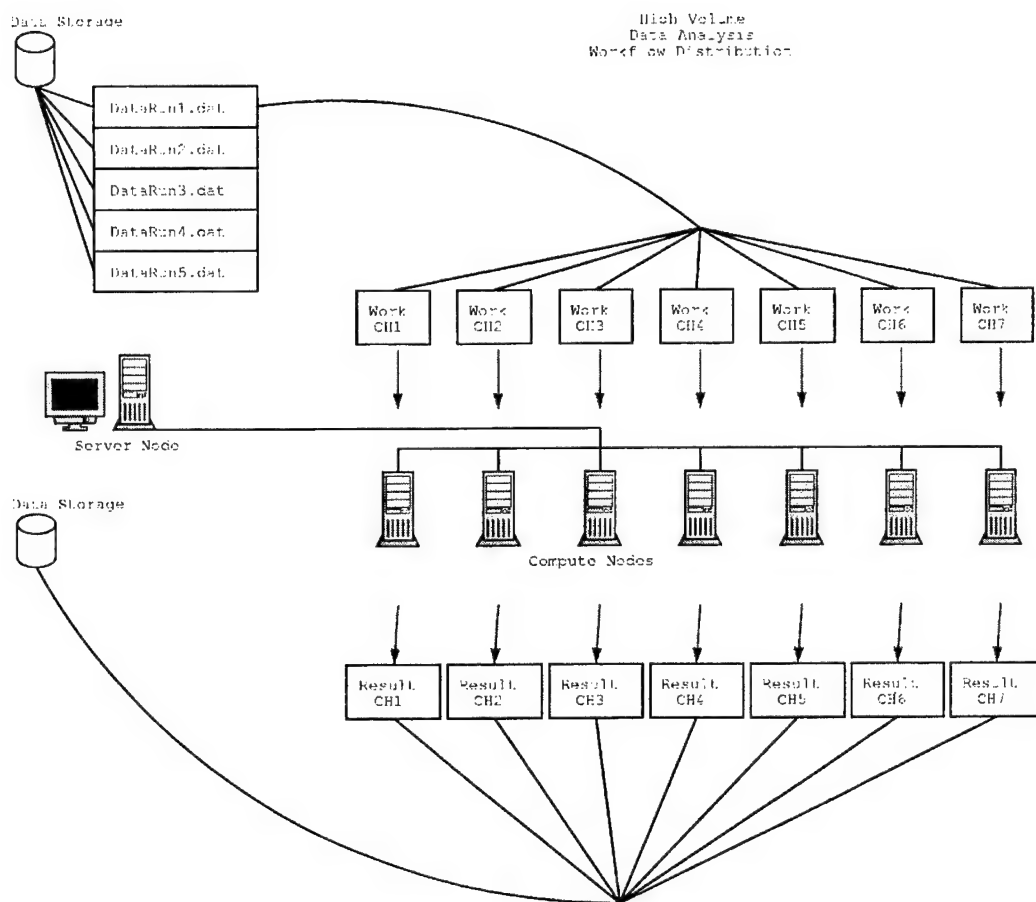
Figure 13: HVDAS Work Flow Distribution

HVDASuite
Server
Algorithm
Overview

Read Command Line Options And Determine Mode Of Operaiton

No — If No Bad Options

No — If MPI Mode

Check Config Files

No — If No Bad Config Files

Build Master File List

No — While There Are Files Left To Be Analyzed

Get The Next File To Be Analyzed

No/Loop — While There Are Channels Left To Be Analyzed

Get The Next Channel To Be Analyzed

Send The Information Needed To Analyze The Next Channel To The First Available Compute Node In The Cluster

MPI Communication VIA Ethernet

Recieve The Results From A Computed Channel Work Request And Store The Results By File, Condition, And Channel

MPI Communication VIA Ethernet

Loop — Store The Results In Memory

Recieve The Outstanding Work Results And Store The Results By File, Condition, And Channel

MPI Communication VIA Ethernet

Summarize and Output Results

Bring Down MPI And Exit

Figure 14: HVDAS Server Algorithm Overview

11.3.2 Client Algorithm Overview

Figure 15 shows how the client portion of HVDAS loops, receives work requests from the server node, performs calculations, and sends results back to the server node.

11.4 Data Distribution

HVDAS does not explicitly send any actual data channels to the client nodes using the Message Passing Interface (MPI). HVDAS sends only the information that the clients need to open up the data files themselves. HVDAS is organized this way to minimize the amount of bandwidth used up by server to client communication. This distributed data importing allows the data to be hosted in a wide variety of places.

The data can be easily placed on a RAID-5 archive machine and shared with the client nodes via Microsoft Windows Network File Sharing (SAMBA) or UNIX Network File System (NFS), but that is just one possibility. The HVDAS method of data import allows the data to be hosted on any cluster-accessible file system. Data can even be hosted on each client node; as long as every other client node has network access to it.

One of the most beneficial results of this design is the ability to use distributed, or parallel, file systems. This allows the cluster nodes, or additional data nodes, to stripe and store data, which will maximize storage space, minimize network bottlenecking, and maximize input output speed.

HVDAS works successfully with Microsoft Windows Network File Sharing (SAMBA), UNIX Network File System (NFS), Parallel Virtual File System (PVFS), and all combinations of the three.

11.5 Bandwidth Utilization

Minimized network communication maximizes performance and expandability. Network saturation can kill the performance benefits of using Linux clusters. HVDAS relies heavily on Linux's and UNIX's ability to cache almost everything without any effort from an application. When a client node reads in a single channel from a data file, the operating system, Linux, will try to cache the entire data file in memory regardless of the source of the data file. This means that data files that get read in from networked file systems will get cached in memory by the client node's operating system. If that client's next work request is a different

HVDASuite
Client
Algorithm
Overview

```
┌─────────────────────────────────────┐
│ Read Command Line OptionsAnd Determine│
│           Mode Of Operaiton           │
└─────────────────────────────────────┘

              If No Bad Options          No

                 If MPI Mode             No

            ┌────────────────────┐
            │  Check Config Files │
            └────────────────────┘

                 If No Bad                No
                 Config Files

          While This Client Has Not Received   No
           The DIE Tag Message To Stop
```

MPI Communication VIA Ethernet

```
              Receive A Message

            If The First Message         No
            Is Not The DIE Tag
```

MPI Communication VIA Ethernet

```
   Receive Every Single Piece Of Information Needed
            To Do Some Amount Of Work
              From The Server Node

              ┌──────────────┐
              │  DoSomeWork   │
              └──────────────┘

Loop
        Send Every Single Piece Of Information
   Needed To Index The Results As Well As The Results
          Themselves Back To The Server Node

                 ╭────────────╮
                 │ Bring Down  │
                 │    MPI      │
                 │  And Exit   │
                 ╰────────────╯
```

MPI Communication VIA Ethernet

Figure 15: HVDAS Client Algorithm Overview

41

channel in the same data file, which will happen quite often if the number of nodes in the cluster is less than the number of channels in the data file, the operating system will read the data from cache instead of reading it again over the network. This behavior does not require any special programming, which is another reason to not send the data channels using MPI.

11.6   Storage Minimization

All of HVDAS' results are stored as ASCII tab-delimited text. Ensuring that all of the results files, with a significant amount of data in them, are not padded with any kind of white space or extra characters reduces file size. HVDAS can optionally skip individual file statistics output or condition summary output to further cut down on results storage space.

11.7   Scalability

HVDAS can scale from one to thousands of nodes. It is possible to run one server and one client portion of HVDAS on a single processor. It is possible to run one server and two client portions of HVDAS on a dual processor computer to use both processors at the same time. It is also possible to run HVDAS on a cluster of single and/or Symmetric Multi-Processing (SMP) nodes and use every processor in the cluster. HVDAS will currently only scale up to the number of channels in your data files. If you try to analyze a set of data files, that contain 100 channels each, on a cluster with 128 processors, HVDAS will only use 101 processors; 100 client processors and 1 server processor.

11.8   Performance Maximization

HVDAS is quite fast even on a single processor computer. Huge linear performance increases come as a result of adding nodes and processors. Network speed, processor speed, memory speed, system bus speed, and input output storage subsystem speed also greatly impact performance. The key to maximizing the parallel calculation performance increase is to make sure that the nodes spend more time calculating than they do communicating or waiting.

HVDAWeibullAnalysis Parallel Benefits -
35 Minutes Of Real Data. 2 Conditions, 20 Data Channels
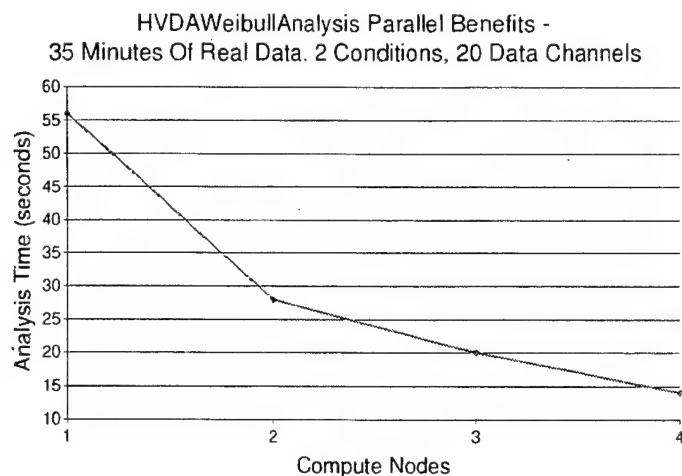
Figure 16: HVDAS Parallel Benefits

11.9 Parallel Benefits

Figure 16 shows the benefits of using HVDAS on a Linux cluster. It is evident that the compute time roughly halves when the number of nodes doubles. The parallel calculation time is nearly $CalcTime_n = CalcTime_1 * \frac{1}{n}$, where n is the number of processors. This behavior will continue until the communication network or some other part of the Linux cluster becomes saturated. This performance increase behavior makes analysis time prediction very easy. Using this behavior, it is possible to predict the analysis time for different data sets or additional processors.

12 Data Formats

This software supports four data formats: ASCII columns, 4-byte binary interleaved with high-byte-first byte ordering, 4-byte binary interleaved with low-byte-first byte ordering, and 2-byte unsigned integer interleaved with low-byte-first byte ordering. The data format is a required input for the program to function properly. If the data are in any other format, they must be converted to one of the acceptable formats prior to being run.

Using the wrong format may not cause a fatal error but would cause the output to be completely meaningless. The headers in all data files are ignored.

If you have a header that exists in your data on a per scan basis, you should try

43

to read the scan header in as one or more dummy channels instead of converting data formats or writing another input plug-in.

12.1 ASCII Columns

In the analysis software, the headers of the ASCII files are skipped. The data are assumed to be in vertical, tab-delimited columns.

12.2 Binary Interleaved High Byte First

- IEEE 32-bit (4-byte) single precision floating point numbers,
- Big endian (high-byte-first) byte ordering,
- Stored using interleaved data points.

12.3 Binary Interleaved Low Byte First

- IEEE 32-bit (4-byte) single precision floating point numbers,
- Little endian (low-byte-first) byte ordering,
- Stored using interleaved data points.

12.4 Binary Interleaved Unsigned Short

- IEEE 16-bit (2-byte) unsigned integers,
- Little endian (low-byte-first) byte ordering,
- Stored using interleaved data points.
- This data type is particularly useful when reading in data stored as integer scans or integer counts.

13 Input

Input to the HVDAS programs can be highly varied, but there are a few guidelines to follow. Keep it as simple as possible. Analyze similar data at the same time. Keep these things in mind when collecting data that you plan on analyzing with HVDAS.

13.1 When HVDAS is Simple to Set Up

Analyze the data from a trial, which are made up of multiple test conditions, as one data set. The simplest input directory structure is organized as /Trial0/Condition0-N/Run0-N, where Runs 0-N are numbered sequentially and follow the convention File Prefix<0-N>File Extension. This is the easiest possible input structure to analyze because HVDAS was built around this input structure. Everything will be easy to set up if the number of channels, the channel order, the scale factors, the sample rate, and the data file format remained constant.

13.2 When HVDAS is Complicated to Set Up

If the input data set is not set up in the ideal structure, all is not lost, but you should be more familiar with running the code to avoid running into problems. Simple UNIX scripts can be written to rename files and folders in bulk, which will make the user's job less painful. If the data exist in multiple file formats or sample rates, you should consider running the analysis in two phases or converting all of the data to a common file format. Configuration will be more difficult to set up if the number of channels, channel order, scale factors, sample, or data file format vary.

13.3 Example Input Structure

This is just one possible file path structure. This example assumes that each condition folder contains binary data files named datarun_0.binary through datarun_3.binary, but each condition does not have to be identical. Example output and configuration files are detailed in their respective sections.

/server/test/condition1/
/server/test/condition2/
/server/test/condition3/
/server/test/HVDAGlobalConfiguration.conf
/server/test/HVDAMMMConfiguration.conf

/server/test/HVDAPSDConfiguration.conf
/server/test/HVDARAOConfiguration.conf
/server/test/HVDAWeibullConfiguration.conf

14 Output

Output from every program in HVDAS is visible from Excel, Star Office, Open Office, Word Pad, gedit, and gnuplot. The results are saved as generic ASCII, tab-delimited, text files. All header information in HVDAS output files have a preceding "#", which makes them compatible with gnuplot.

Output is stored in a directory structure similar to the input data structure. The base results folder, named by the user in a 'HVDA*Configuration.conf' file, will contain all of the output. For basic MMM analysis, the results will be in the sub-folder named 'HVDAMMMResults'. Modified and converted input data will be stored in the sub-folder named 'HVDAConvertedOutputData'. For PSD analysis, the results will be in the sub-folder named 'HVDAPSDResults'. For RAO analysis, the results will be in the sub-folder named 'HVDARAOResults'. For Weibull analysis, the results will be in the sub-folder named 'HVDAWeibullResults'. At the next level the results are broken down into their respective conditions using the condition labels defined in the 'HVDAGlobalConfiguration.conf' file. These folders contain the actual condition results, sorted by filter type, original data file name and index.

14.1 MMM Analysis Output

Each row in the MMM output represents a channel of analyzed data and contains the Channel Number, Channel Name, Mean, Max, Min, $(Maximum - Mean)$, $(Minimum - Mean)$, Range, Maximum Peak-To-Peak, Standard Deviation, Kurtosis, Mean Crossings, and the ratio $\frac{(Maximum-Mean)}{(StandardDeviation)}$. Other pertinent information is in the header at the top of the output file. Each condition, or folder, is summarized on a channel-by-channel basis. The summary gives values for each channel for each statistical value: Mean, $(Maximum - Mean)$, etc.

For each condition, the program will create one results file for each input file for each of four possible filter states. In addition, one summary file is created for each filter state in each condition and is saved in the results folder for that condition. As with individual files, pertinent information is written at the top of the summary file.

46

Output is stored in a directory structure similar to the input data structure. The base results folder, named by the user in the 'HVDAMMMConfiguration.conf' file, will contain all of the output. For basic MMM analysis the results will be in the sub-folder named 'HVDAMMMResults'. Modified and converted input data will be stored in the sub-folder named 'HVDAConvertedOutputData'. At the next level the results are broken down into their respective conditions using the condition labels defined in the 'HVDAGlobalConfiguration.conf' file. These folders contain the actual condition results, sorted by filter type, original data file name and index.

14.1.1 Example Output Structure

This is just one possible output structure. Your actual output structure will depend on your choices for base folder name, condition labels, desired filters, and the number of files analyzed. The individual and summary files are shown below.

```
/server/test/HVDAMMMResults/condition1/HVDAMMM_Raw_condition1_datarun_0.txt
/server/test/HVDAMMMResults/condition1/HVDAMMM_Raw_condition1_datarun_1.txt
/server/test/HVDAMMMResults/condition1/HVDAMMM_Raw_condition1_datarun_2.txt
/server/test/HVDAMMMResults/condition1/HVDAMMM_Raw_condition1_datarun_3.txt
/server/test/HVDAMMMResults/condition1/HVDAMMM_Raw_condition1_Summary.txt
/server/test/HVDAMMMResults/condition2/HVDAMMM_Raw_condition2_datarun_0.txt
/server/test/HVDAMMMResults/condition2/HVDAMMM_Raw_condition2_datarun_1.txt
/server/test/HVDAMMMResults/condition2/HVDAMMM_Raw_condition2_datarun_2.txt
/server/test/HVDAMMMResults/condition2/HVDAMMM_Raw_condition2_datarun_3.txt
/server/test/HVDAMMMResults/condition2/HVDAMMM_Raw_condition2_Summary.txt
/server/test/HVDAMMMResults/condition3/HVDAMMM_Raw_condition3_datarun_0.txt
/server/test/HVDAMMMResults/condition3/HVDAMMM_Raw_condition3_datarun_1.txt
/server/test/HVDAMMMResults/condition3/HVDAMMM_Raw_condition3_datarun_2.txt
/server/test/HVDAMMMResults/condition3/HVDAMMM_Raw_condition3_datarun_3.txt
/server/test/HVDAMMMResults/condition3/HVDAMMM_Raw_condition3_Summary.txt
```

14.1.2 Example Output File

Figure 17 is an example MMM output summary file. This graphic was printed to a file from Open Office Calc.

Sheet1

#HVDA MMM Analysis Weighted Summary
#HVDA Library Version: 2003-10-23
#Date: Fri Oct 24 09:55:26 2003
#User: code653
#Hostname: fnrhis
#Filter State: Raw Data
#Condition Label: 1999102208163 9
#Scale Factors: /HVDAGlobalConfiguration.conf
#Summary File: /home/code653/CRSData/hvdas/HVDAMMMResults/1999102208163 9/1999102208163 9_HVDAMMM__Raw_Summary.txt
#Sample Rate: 5.00000e+00 Samples Per Second
#Total Run Time: 1800.200 Seconds (Calculated Using Sample Rate)
#Data File Type: BINARY_INTERLEAVED_LOW_BYTE_FIRST
#Number Of Files: 1 Total Files In This Condition
#Sea State: 4
#Speed: 19.7
#Heading: 344

#Ch # Channel Name	Mean	Max	Min	Max-Mean	Min-Mean	Range	Max PeakToPeak	Standard Dev	Kurtosis	Mean Crossings	Max-Mean/Stdev
0 Roll	4.16E-001	1.24E+000	-1.21E+000	8.26E-001	-1.63E+000	2.45E+000	2.26E+000	3.95E-001	4.83	85	2.09
1 Pitch	-4.58E-001	-2.76E-001	-6.40E-001	1.82E-001	-1.81E-001	3.64E-001	3.60E-001	5.33E-002	3.29	320	3.42
2 Heave	3.56E-001	8.07E-001	8.71E-002	4.51E-001	-2.69E-001	7.20E-001	6.83E-001	1.09E-001	4.46	216	4.14
3 VerticalBowDisp	2.66E-002	5.30E-001	-4.80E-001	5.03E-001	-5.07E-001	1.01E+000	1.01E+000	1.33E-001	3.78	494	3.8
4 VBM frame 256	2.98E-004	2.16E+002	-2.54E+002	2.16E+002	-2.54E+002	4.70E+002	3.94E+002	5.31E+001	3.41	1532	4.06
5 VBM frame 274	8.61E-004	2.53E+002	-2.95E+002	2.53E+002	-2.95E+002	5.47E+002	4.54E+002	6.64E+001	3.11	1618	3.8
6 VBM frame 324	9.32E-005	1.63E+002	-2.05E+002	1.63E+002	-2.05E+002	3.68E+002	3.13E+002	4.27E+001	3.15	1550	3.83
7 VBM frame 256 LP	2.31E-003	1.11E+002	-1.39E+002	1.11E+002	-1.39E+002	2.50E+002	2.01E+002	3.57E+001	3.12	524	3.12
8 VBM frame 274 LP	2.10E-003	1.41E+002	-1.66E+002	1.41E+002	-1.66E+002	3.08E+002	2.36E+002	4.27E+001	3.11	570	3.31
9 VMB frame 324 LP	7.99E-004	9.46E+001	-1.22E+002	9.46E+001	-1.22E+002	2.17E+002	1.68E+002	2.98E+001	3.06	618	3.18
10 VBM frame 256 HP	-2.09E-003	1.84E+002	-1.94E+002	1.84E+002	-1.94E+002	3.77E+002	3.69E+002	3.94E+001	4.44	2194	4.67
11 VBM frame 274 HP	-1.30E-003	2.15E+002	-2.22E+002	2.15E+002	-2.22E+002	4.37E+002	4.34E+002	5.09E+001	3.78	2180	4.23
12 VMB frame 324 HP	-6.91E-004	1.40E+002	-1.40E+002	1.40E+002	-1.40E+002	2.79E+002	2.79E+002	3.06E+001	3.88	2262	4.57
13 Rel Wave Height TSK	7.24E-002	1.41E+000	-1.26E+000	1.34E+000	-1.33E+000	2.67E+000	1.94E+000	3.95E-001	2.99	965	3.39
14 Corrected Wave Height TSK	4.59E-003	8.70E-001	-8.30E-001	8.65E-001	-8.35E-001	1.70E+000	1.44E+000	2.37E-001	3	1649	3.66

Page 1

48

Figure 17: Example MMM Output File

14.2   PSD Analysis Output

The first and left most column contains the frequencies that correspond row by row with the PSD magnitudes. Each other column represents a channel of analyzed data and contains the Channel Name and PSD, which was computed using the PSDFFTMSAM. Other pertinent information is in the header at the top of the output file, including the parameters used to calculate the PSD I-Skip and Segment Length. Each condition, or folder, is summarized on a channel-by-channel basis. The summary gives values for each frequency's PSD magnitude for each channel.

For each condition, the program will create one results file for each input file for either raw or low-pass filter states. In addition, one summary file is created for either the raw or low-pass filter state in each condition and is saved in the results folder for that condition. As with individual files, pertinent information is written at the top of the summary file.

Output is stored in a directory structure similar to the input data structure. The base results folder, named by the user in the 'HVDAPSDConfiguration.conf' file, will contain all of the output. For this basic analysis, the results will be in the sub-folder named 'HVDAPSDResults'. At the next level, the results are broken down into their respective conditions using the condition labels defined in the 'HVDAGlobalConfiguration.conf' file. These folders contain the actual condition results, sorted by filter type, original data file name and index.

14.2.1   Example Output Structure

This is just one possible output structure. Your actual output structure will depend on your choices for base folder name, condition labels, desired filters, and the number of files analyzed. The individual and summary files are shown below.

    /server/test/HVDAPSDResults/condition1/HVDAPSD_Raw_condition1_datarun_0.txt
    /server/test/HVDAPSDResults/condition1/HVDAPSD_Raw_condition1_datarun_1.txt
    /server/test/HVDAPSDResults/condition1/HVDAPSD_Raw_condition1_datarun_2.txt
    /server/test/HVDAPSDResults/condition1/HVDAPSD_Raw_condition1_datarun_3.txt
    /server/test/HVDAPSDResults/condition1/HVDAPSD_Raw_condition1_Summary.txt
    /server/test/HVDAPSDResults/condition2/HVDAPSD_Raw_condition2_datarun_0.txt
    /server/test/HVDAPSDResults/condition2/HVDAPSD_Raw_condition2_datarun_1.txt
    /server/test/HVDAPSDResults/condition2/HVDAPSD_Raw_condition2_datarun_2.txt
    /server/test/HVDAPSDResults/condition2/HVDAPSD_Raw_condition2_datarun_3.txt
    /server/test/HVDAPSDResults/condition2/HVDAPSD_Raw_condition2_Summary.txt
    /server/test/HVDAPSDResults/condition3/HVDAPSD_Raw_condition3_datarun_0.txt

49

/server/test/HVDAPSDResults/condition3/HVDAPSD_Raw_condition3_datarun_1.txt
/server/test/HVDAPSDResults/condition3/HVDAPSD_Raw_condition3_datarun_2.txt
/server/test/HVDAPSDResults/condition3/HVDAPSD_Raw_condition3_datarun_3.txt
/server/test/HVDAPSDResults/condition3/HVDAPSD_Raw_condition3_Summary.txt

14.2.2 Example Output File

Figure 18 is an example PSD output summary file. This graphic was printed to a file from Open Office Calc.

14.3 RAO Analysis Output

The first and left most column contains the frequencies that correspond row by row with the root RAO magnitudes. Each other column represents a channel of analyzed data and contains the Channel Name and square root of the RAO magnitude, which was computed using $\sqrt{RAO} = \sqrt{\frac{PSDFFTMSAM(DataChannel)}{PSDFFTMSAM(DivisorChannel)}}$. Other pertinent information is in the header at the top of the output file, including the parameters used to calculate the PSD I-Skip and Segment Length. Each condition, or folder, is summarized on a channel-by-channel basis. The summary gives values for each frequency's PSD magnitude for each channel.

For each condition, the program will create one results file for each input file for either raw or low-pass filter states. In addition, one summary file is created for either the raw or low-pass filter state in each condition and is saved in the results folder for that condition. As with individual files, pertinent information is written at the top of the summary file.

Output is stored in a directory structure similar to the input data structure. The base results folder, named by the user in the 'HVDARAOConfiguration.conf' file, will contain all of the output. For this basic analysis the results will be in the sub-folder named 'HVDARAOResults'. At the next level the results are broken down into their respective conditions using the condition labels defined in the 'HVDAGlobalConfiguration.conf' file. These folders contain the actual condition results, sorted by filter type, original data file name and index.

14.3.1 Example Output Structure

This is just one possible output structure. Your actual output structure will depend on your choices for base folder name, condition labels, desired filters, and the number of files analyzed. The individual and summary files are shown below.

Sheet1

#HVDA PSD Analysis
#HVDALibrary Version: 2003-11-06
#Date: Thu Nov 13 14:43:30 2003
#User: code653
#Hostname: fixthis
#Filter State: Raw Data
#Condition Label: 20000120083250
#Data File: /home/code653/CRSData/data/individual_runs/20000120083250/20000120083250.dtb
#Scale Factors: HVDAGlobalConfiguration-reruns.conf
#Results File: /home/code653/CRSData/hvdas/HVDAPSDResults/20000120083250/20000120083250_HVDAPSD_Raw_20000120083250.txt
#Sample Rate: 5 Samples Per Second
#Highest Frequency Of Interest: 0.5
#Over Sampling: 10
#Target Delta Frequency: 0.01
#Data File Type: BINARY_INTERLEAVED_LOW_BYTE_FIRST
#Data Size: 4
#Sea State: 3
#Speed: 23
#Heading: 140

#Encounter Frequency	Wave Frequency	Ch 7 VBM frame 256 LP	Ch 8 VBM frame 274 LP	Ch 9 VMB frame 324 LP
#Hz	Hz			
0.00E+000	-1.00E+000	1.64E-012	7.61E-013	1.14E-012
1.00E-002	-1.00E+000	7.83E+000	6.09E+001	1.62E+001
2.00E-002	-1.00E+000	8.42E+000	1.69E+001	1.34E+001
3.00E-002	-1.00E+000	1.31E+001	2.48E+001	1.82E+001
4.00E-002	-1.00E+000	8.34E+000	1.93E+001	1.51E+001
5.00E-002	-1.00E+000	1.07E+001	1.77E+001	1.23E+001
6.00E-002	-1.00E+000	1.02E+001	1.75E+001	1.68E+001
7.00E-002	-1.00E+000	1.78E+001	2.63E+001	2.08E+001
8.00E-002	-1.00E+000	3.04E+001	5.43E+001	4.21E+001

Page 1

51

Figure 18: Example PSD Output File

```
/server/test/HVDARAOResults/condition1/HVDARAO_Raw_condition1_datarun_0.txt
/server/test/HVDARAOResults/condition1/HVDARAO_Raw_condition1_datarun_1.txt
/server/test/HVDARAOResults/condition1/HVDARAO_Raw_condition1_datarun_2.txt
/server/test/HVDARAOResults/condition1/HVDARAO_Raw_condition1_datarun_3.txt
/server/test/HVDARAOResults/condition1/HVDARAO_Raw_condition1_Summary.txt
/server/test/HVDARAOResults/condition2/HVDARAO_Raw_condition2_datarun_0.txt
/server/test/HVDARAOResults/condition2/HVDARAO_Raw_condition2_datarun_1.txt
/server/test/HVDARAOResults/condition2/HVDARAO_Raw_condition2_datarun_2.txt
/server/test/HVDARAOResults/condition2/HVDARAO_Raw_condition2_datarun_3.txt
/server/test/HVDARAOResults/condition2/HVDARAO_Raw_condition2_Summary.txt
/server/test/HVDARAOResults/condition3/HVDARAO_Raw_condition3_datarun_0.txt
/server/test/HVDARAOResults/condition3/HVDARAO_Raw_condition3_datarun_1.txt
/server/test/HVDARAOResults/condition3/HVDARAO_Raw_condition3_datarun_2.txt
/server/test/HVDARAOResults/condition3/HVDARAO_Raw_condition3_datarun_3.txt
/server/test/HVDARAOResults/condition3/HVDARAO_Raw_condition3_Summary.txt
```

14.3.2 Example Output File

Figure 19 is an example RAO output summary file. This graphic was printed to a file from Open Office Calc.

14.4 Weibull Analysis Output

The Weibull output is organized by channels. Each input channel yields up to 5 output columns depending on the calculation performed. The result columns report the actual data amplitudes used, linearized Weibull X data, linearized Weibull Y data, and linear Weibull trend lines.

Other pertinent information is in the header at the top of the output file, including the parameters used to calculate the Weibull lifetime statistics. Each condition, or folder, is summarized on a channel-by-channel basis. For each condition, the program will create one results file for either raw, low-pass, or high-pass filter states.

Output is stored in a directory structure similar to the input data structure. The base results folder, named by the user in the 'HVDAWeibullConfiguration.conf' file, will contain all of the output. For this basic analysis the results will be in the sub-folder named 'HVDAWeibullResults'. At the next level the results are broken down into their respective conditions using the condition labels defined in the

Sheet1

#HVDA RAO Analysis
#HVDALibrary Version:
#Date: 2003-11-06
#User: Thu Nov 13 14:43:46 2003
#Hostname: code653
#Filter State: fixthis
#Condition Label: Raw Data
#Data File: 20000120083250
#Scale Factors: /home/code653/CRSData/data/individual_runs/20000120083250/20000120083250.dtb
#Results File: HVDAGlobalConfiguration-reruns.conf
#Sample Rate: /home/code653/CRSData/hvdas/HVDARAOResults/20000120083250/20000120083250_HVDARAO_Raw_20000120083250.txt
#Highest Frequency Of Interest: 5 Samples Per Second
#Over Sampling: 0.5
#Target Delta Frequency: 10
#Data File Type: 0.01
#Data Size: BINARY_INTERLEAVED_LOW_BYTE_FIRST
#Sea State: 4
#Speed: 3
#Heading: 23
140

| #Encounter Frequency | Wave Frequency | Ch 7 | Ch 8 | Ch 9 |
#Hz	Hz	VBM frame 256 LP	VBM frame 274 LP	VMB frame 324 LP
0.00E+000	-1.00E+000	0.00E+000	0.00E+000	0.00E+000
1.00E-002	-1.00E+000	6.96E-001	1.94E+000	1.00E+000
2.00E-002	-1.00E+000	7.94E-001	1.13E+000	1.00E+000
3.00E-002	-1.00E+000	8.49E-001	1.17E+000	1.00E+000
4.00E-002	-1.00E+000	7.44E-001	1.13E+000	1.00E+000
5.00E-002	-1.00E+000	9.33E-001	1.20E+000	1.00E+000
6.00E-002	-1.00E+000	7.79E-001	1.02E+000	1.00E+000
7.00E-002	-1.00E+000	9.26E-001	1.13E+000	1.00E+000
8.00E-002	-1.00E+000	8.50E-001	1.14E+000	1.00E+000

Page 1

53

Figure 19: Example RAO Output File

'HVDAGlobalConfiguration.conf' file. These folders contain the actual condition results, sorted by filter type, original data file name and index.

14.4.1 Example Output Structure

This is just one possible output structure. Your actual output structure will depend on your choices for base folder name, condition labels, desired filters, and the number of files analyzed. The individual and summary files are shown below.

    /server/test/HVDAWeibullResults/condition1/HVDAWeibull_Raw_condition1_Summary.txt
    /server/test/HVDAWeibullResults/condition2/HVDAWeibull_Raw_condition2_Summary.txt
    /server/test/HVDAWeibullResults/condition3/HVDAWeibull_Raw_condition3_Summary.txt

14.4.2 Example Output File

Figure 20 is an example Weibull output summary file. This graphic was printed to a file from Open Office Calc.

15 Configuration Files and Options

HVDAS has the ability to generate example configuration files; just run "hvdas - createexampleconfigs". The example configuration files are heavily documented. Basic documentation is provided in this document, while more extensive documentation and examples are also provided in the example configuration files. Every HVDAS configuration file is a lot like a standard UNIX configuration file. They all end with the extension ".conf" and can have comment lines that start with "#". If you choose to generate example configuration files, existing configuration files will be overwritten without warning.

15.1 HVDAGlobalConfiguration.conf

The Global Configuration file is a single file containing parameters and settings that define the entire data set. The global configuration values are as follows.

15.1.1 Data Type

This is a setting that describes the data storage format. A full description of the data types exists in the example configuration files and in the Data Formats section.

Sheet1

#HVDA Weibull Analysis Summary

#Date:	Wed Oct 29 10:29:43 2003
#User:	code653
#Hostname:	fxthis
#Filter State:	Raw or Already Filtered Data
#Condition Label:	19991022081639
#Scale Factors:	HVDAGlobalConfiguration.conf
#Summary File:	/home/code653/CRSData/hvdas/HVDAWeibullResult/19991022081639/19991022081639_HVDAWeibull_2ParameterMomentMethod_Maximums-Mean_Raw_Summary.txt
#Sample Rate:	5 Samples Per Second
#Ship Lifetime:	20
#Extreme F:	0.95
#Condition Probability:	0.01
#BetaLowerBound:	0.2
#BetaUpperBound:	10
#BetaStepSize:	0
#WeibullType:	2 Parameter Moment Method
#AmplitudeType:	Maximums-Mean
#Data File Type:	BINARY_INTERLEAVED_LOW_BYTE_FIRST
#Number Of Files:	1
#RunTimeInSeconds:	1800.2
#Sea State:	4
#Speed:	19.7
#Heading:	344
#PredictionTimeInSeconds:	0

#ChannelNumber		7	8
#ChannelName		VBM frame 256 LP	VBM frame 274 LP
#WeibullSize		262	285
#LinearRegressionSlope		1.36E+000	1.13E+000
#LinearRegressionYIntercept		-4.91E+000	-4.34E+000
#LinearRegressionCorrelationCoefficient		9.98E-001	9.85E-001
#LinearRegressionCharacteristicValue		3.75E+001	4.70E+001
#LinearRegressionParentWeibullLifetimeMax		2.57E+002	4.83E+002
#LinearRegressionExtremeValueAnalysisOfParentWeibull		2.97E+002	5.74E+002
#MomentMethodSlope		1.51E+000	1.54E+000
#MomentMethodYIntercept		-5.47E+000	-5.91E+000
#MomentMethodCharValue		3.74E+001	4.57E+001
#MomentMethodParentWeibullLifetimeMax		2.12E-002	2.53E+002
#MomentMethodExtremeValueAnalysisOfParentWeibull		2.41E+002	2.87E+002
#AmplitudeMean		3.37E+001	4.11E+001
#AmplitudeMax		1.11E+002	1.41E+002
#AmplitudeMin		9.72E-001	2.58E-001
#AmplitudeVariance		5.18E+002	7.47E+002
#AmplitudeKurtosis		3.21E+000	3.10E+000
#MomentMethodXNot		0.00E+000	0.00E+000
#LinearRegressionXNot		0.00E+000	0.00E+000
#UnsortedAmplitudes			

#LinearWeibullXData	LinearWeibullYData	LinearRegressionXData	LinearRegressionYData	MomentMethodLinearXData	MomentMethodLinearYData	#UnsortedAmplitudes	LinearWeibullXData	LinearWeibullYData	LinearRegressionXData	LinearRegressionYData	MomentMethodLinearYData
6.49E-001	-2.81E-002	-5.37E+000	-4.97E+000	-5.52E+000		7.28E+001	-1.36E+000	-5.65E+000	-5.87E+000	-7.99E+000	
4.69E-001	1.91E-001	-4.88E+000	-4.67E+000	-5.18E+000		6.07E+001	-1.01E+000	-4.96E+000	-5.47E+000	-7.45E+000	
3.59E-001	3.17E-001	-4.47E+000	-4.50E+000	-4.99E+000		4.03E+001	-6.60E-001	-4.55E+000	-5.08E+000	-6.92E+000	
5.12E-001	4.67E-001	-4.13E+000	-4.30E+000	-4.77E+000		6.06E+001	-1.29E-001	-4.26E+000	-4.49E+000	-6.11E+000	
4.96E-001	6.02E-001	-3.95E+000	-4.11E+000	-4.56E+000		6.58E+001	1.82E-002	-4.04E+000	-4.32E+000	-5.88E+000	

55

Figure 20: Example Weibull Output File

Keyword: DATA_TYPE

This setting is spelling and case sensitive, so the options must be input EX-ACTLY as listed below. Choose only one.

- ASCII_COLUMNS: ASCII data stored in interleaved format (columns are channels and rows are samples). Usually tab delimited.

- BINARY_INTERLEAVED_HIGH_BYTE_FIRST: Binary real data stored in interleaved format with the high byte first.

- BINARY_INTERLEAVED_LOW_BYTE_FIRST: Binary real data stored in interleaved format with the low byte first.

- BINARY_INTERLEAVED_UNSIGNED_SHORT: Binary integer data stored in interleaved format with the low byte first.

15.1.2 Data Size

This is the number of bytes that each data point or value takes up. The default value is fixed at 4 bytes, a standard single precision floating point number. Future versions will allow for higher precision. This setting has no effect for ASCII data.

Keyword: DATA_SIZE

15.1.3 Sample Rate

The number of samples recorded per second per channel during acquisition. This number can be scaled up or down to aid in the conversion from model scale to full scale or vice verse. If you wish to scale the results, remember to change every single value to the desired scale. You would need to convert frequencies, scale factors, and speeds in multiple configuration files to the same scale.

Keyword: SAMPLE_RATE

15.1.4 Total Number of Acquired Channels

This number must include dead and/or dummy channels that were saved during acquisition. These channels may be ignored later.

Keyword: NUMBER_OF_CHANNELS

15.1.5 Header Size

The header size is an integer representing the number of bytes for a binary file header or the number of lines for an ASCII file header.
Keyword: HEADER_SIZE

15.1.6 Drift Compensation Seconds

The drift compensation seconds parameter is used to compensate for non-stationary mean shift behavior that you wish to eliminate from your data after the data has been read in and scaled. A value < 0 will not compensate, a value of 0 will subtract the entire time history's mean, and a value > 0 will subtract the local mean for the desired number of seconds.
Keyword: DRIFT_COMPENSATION_SECONDS

15.1.7 Total Number of Acquired Channels

This number must include dead and/or dummy channels that were saved during acquisition. These channels may be ignored later.
Keyword: NUMBER_OF_CHANNELS

15.1.8 Header Size

The header size is an integer representing the number of bytes for a binary file header or the number of lines for an ASCII file header.
Keyword: HEADER_SIZE

15.1.9 Test Condition List

This is a list containing the paths to each test condition directory to be analyzed in the entire data block as well as information describing each test condition. Each entry must be on a separate line and contain every piece of information. These entries are case sensitive. If the entered directory is not found during a self-test, the directory will not be analyzed or an error will occur depending on the problem with the directory path. As with all configuration lists, the test condition list must end with 'END' on a separate line after the last test condition entry. The user must have the proper permissions to access the data directories and data files before

running the program or it will terminate in error. The test condition list is a tab-delimited matrix of information describing all of the test conditions, which lends itself to being edited with a spreadsheet application.

Keyword: TEST_CONDITION_LIST

This list contains the following parameters to describe each test condition.

CONDITION_LABEL
FILE_PREFIX
FILE_EXTENSION
FILE_ID_LOWER_BOUND
FILE_ID_UPPER_BOUND
CONDITION_START_SKIP_TIME
CONDITION_END_KEEP_TIME
SEA_STATE
SPEED_IN_KNOTS
HEADING_IN_DEGREES
CONDITION_PROBABILITY

15.1.10 Lowest and Highest Data File Indexes

The lowest and highest number following the file prefix in the file names of the data files in the condition directory.

Keyword: FILE_ID_LOWER_BOUND: FILE_ID_UPPER_BOUND

15.1.11 Data File Prefix

The part of the file name after the directory path and before the indexing number, include any characters before beginning the number including spaces and special characters.

Keyword: FILE_PREFIX

15.1.12 Data File Extension

The file extension of the data files in the corresponding condition folder.

Keyword: FILE_EXTENSION

15.1.13 Condition Label

The user defined 'name' for the condition. This has no bearing on the actual analysis, but the program will terminate if two folders have the exact same condition label.

Keyword: CONDITION_LABEL

15.1.14 Condition Probability

The user defined probability for the condition as a percentage of the ship's lifetime. This value is used in Weibull estimation of maximum lifetime statistics.

Keyword: CONDITION_PROBABILITY

15.1.15 Sea State

This sets the sea state of the data in the current condition. For example, in analyzing ship structure data collected in a Sea State 7, set this value to 7 or SS7. If you are not analyzing ship structure data, you may enter an identifier that will define the type of test or condition you are analyzing. The combination of sea state, speed, and heading will be used to generate trend tables for all parts of HVDAS.

Keyword: SEA_STATE

15.1.16 Speed In Knots

This sets the speed of the data in the current condition. For example, in analyzing ship structure data collected at 15 Knots, set this value to 15. Make sure that your speed matches your chosen scale. Do everything in either model or full scale for ship structures. The default value is 0.0. The speed is used by RAO analysis in the conversion from encounter to wave frequency.

Keyword: SPEED_IN_KNOTS

15.1.17 Heading in Degrees

This setting sets the heading of the data in the current condition in degrees. For example, 0.0 degrees is assumed to mean that the ship is heading straight into the waves, which is known as "head seas". The default value is 0.0. The heading is used by RAO analysis in the conversion from encounter to wave frequency.

Keyword: HEADING_IN_DEGREES

15.1.18   Condition Start Skip Time Selection

This field represents the amount of time you would like to skip at the beginning of a condition; in seconds. This feature is very useful if you have data files that were collected in even intervals. For example, the RV *Triton* data conditions are comprised of data files that were recorded in 5 minute increments. The condition you may want to analyze might require you to skip the first 2 minutes of the first data file and keep the first 4 minutes of the last data file. This setting allows you to do that. Example: 'CONDITION_START_SKIP_TIME:\t60' to skip the first minute of the first data file in the condition. Leave the default value of 0.0 if you would like to analyze the entire condition worth of data. Note: In order to use this feature, you must set FILE_ID_UPPER&LOWER to the first and last files in the current condition respectively.

Key Word: CONDITION_START_SKIP_TIME

15.1.19   Condition End Keep Time Selection

This field represents the amount of time you would like to keep at the end of a condition; in seconds. This feature is very useful if you have data files that were collected in even intervals. For example, the RV *Triton* data conditions are comprised of data files that were recorded in 5 minute increments. The condition you may want to analyze might require you to skip the first 2 minutes of the first data file and keep the first 4 minutes of the last data file. This setting allows you to do exactly that. Example: 'CONDITION_END_KEEP_TIME:\t240' to keep the first four minutes and truncate the last minute of the last data file in the condition. Leave the default value of 0.0 if you would like to analyze the entire condition worth of data. Note: In order to use this feature, you must set FILE_ID_UPPER&LOWER to the first and last files in the current condition respectively. If you have a condition that contains only one data file, set the FILE_IDs to the same file, then the skip time will be applied followed immediately by the end keep time; in this case the end keep time should be the total time occupied by the condition or 0.0 to use all of it automatically.

Key Word: CONDITION_END_KEEP_TIME

15.1.20   Channel List

The channel list is a tab-delimited list of channel names, scale factors, and zero values. ALL channel names must be listed, even if the channel is empty, dead or

```
DATA_TYPE                      BINARY_INTERLEAVED_HIGH_BYTE_FIRST
DATA_SIZE                      4
SAMPLE_RATE                    200
NUMBER_OF_CHANNELS             1
HEADER_SIZE                    8
DRIFT_COMPENSATION_SECONDS    -1

TEST_CONDITION_LIST   CONDITION_LABEL   FILE_PREFIX   FILE_EXTENSION   FILE_ID_LOWER_BOUND   FILE_ID_UPPER_BOUND   CONDITION_START_SRIP_TIME   CONDITION_END_KEEP_TIME   SEA_STATE   SPEED_IN_KNOTS   HEADING_IN_DEGREES   CONDITION_PROBABILITY
/server/test/condtest1/   condition1   datarun_   binary   0   3   0   0.553   10   0   0.01
/server/test/condtest2/   condition2   datarun_   binary   0   3   0   0.554   10   0   0.01
/server/test/condtest3/   condition3   datarun_   binary   0   3   0   0.555   10   0   0.01
END

CHANNEL_LIST      SCALE_FACTORS   ZERO_VALUES
Channel1Lines     1               0
Channel2Lines     1               0
Channel3Lines     1               0
END

LOW_PASS_COEFFICIENTS:
   4.95E-005
   2.33E-005
   2.35E-005
END

HIGH_PASS_COEFFICIENTS:
   0.05
   0.02
   0.02
   0.02
END

SMOOTHED_COEFFICIENTS:
  -1.93E-005
   5.91E-005
   3.28E-005
   5.56E-005
   5.91E-005
END
```

Sheet1

Page 1

62

Figure 21: Example HVDAGlobalConfiguration.conf

15.2   HVDAMMMConfiguration.conf

The HVDAMMMConfiguration.conf file contains the parameters and settings specific to the basic MMM statistical analysis. The settings in this file are constant for the entire data block. Therefore, there need be only one copy of this file in the base directory of the analysis software. HVDAMMMConfiguration.conf includes the following parameters.

15.2.1   List of Channels to Analyze

A number in this list refers to a channel in your data file. Channel '0' (zero) refers to the first saved channel. The number of channels minus one references the last saved channel. So, if there are 16 channels, the first channel is 0 and the last is 15. The numbers can be listed in any order as long as the requested channel is less than the total number of channels, greater than or equal to 0, and exists in all data files in all conditions you wish to analyze.

Keyword: CHANNELS_TO_RUN

15.2.2   List of Filter States to Apply

The filter states are referred to using a list of integers.

- 0 - No Filtering, use raw data

- 1 - Smooth-Pass FIR Filter the raw data using the Smoothing Coefficients to remove noise

- 2 - Low-Pass FIR Filter using the Low-Pass Coefficients to remove high-frequency energy

- 3 - High-Pass FIR Filter using the High-Pass Coefficients to remove low-frequency energy

- 4 - Box Car Average smoothing of the data to remove drop-outs and noise

- 5 - Initial Conditions Zero Subtraction subtracts user defined values from each channel for every filter state requested, which means this option must be used in conjunction with at least one other filter state

- 6 - Compute Virtual Channel Expressions from raw input data before converting data files; this option must be used in conjunction with at least one other filter state and can only be used when converting data.

Use one or all of these by putting one integer per line in the configuration file. Each filter type will produce one spreadsheet of output per file and a summary of the filter condition. End the list of filters with 'END'.
  Keyword: FILTERS

15.2.3 Box Car Average Window Size

This is used to select the number of data points on each side of the box car. This value can be even or odd, it does not matter.
  Keyword: BOX_CAR_WINDOW_SIZE

15.2.4 Virtual Channel Expressions

Key Word: 'VIRTUAL_CHANNEL_EXPRESSIONS'
  This list should contain a list of expressions you want to use to compute virtual channels. These virtual channel expressions use postfix notation to eliminate the need to figure out order of operations and to eliminate the need for parenthesis. Postfix notation allows the user to completely control the execution of the expression without parenthesis or brackets. It is used because of the simplicity of implementation. Virtual Channels will only be computed when converting the data using –convertdata to output ASCII data files. The new converted ASCII data files can then be run through the MMM analysis software by changing data file descriptions and running hvdas in MMM MPI mode just like any other data files.
  Virtual Channel Support is not heavily tested and is extremely picky. You must follow the expression spec EXACTLY. You cannot perform expressions on constants. Be sure to do a test channel to make sure that your expression is correct. Every complex expression is made up of simple expressions.
  ////////////////////////////////////
  //expression format spec from the source code
  //operand d, which can be a constant(1, 1.5, 45.987, ...)
  //or a channel (c0, c1, c2,...)
  //operator r, which can be (+, -, *, /, and ^ for simple expressions)
  //or (+, -, *, /, and sqrt) for complex expressions
  //space s

64

//tab t

//comma ,

//semicolon ;

////////////////////////////

//Simple Expression

//d,sd,sd,tr,sr

//or

//Complex Expression composed of Simple Expressions

//d,sd,sd,tr,sr;sd,sd,sd,tr,sr;tEOsr

////////////////////////////

Example simple expression $(c0 + c1 + c2)$ should be input as

'c2, c1, c0,<tab>+, +'

Example simple expression $(c0 + 2 + 1.1)$ should be input as

'1.1, 2, c0,<tab>+, +'

Example simple expression $\frac{c2 \times 5.5}{2.2}$ should be input as

'2.2, 5.5, c2,<tab>*, /'

Example complex expression $(c11 + 3.3) \times (c9 + 3.3)$ should be input as

'3.3, c9,<tab>+; 3.3, c11,<tab>+;<tab>EO *'

Example complex expression $\sqrt{\frac{c2 \times c1}{c4 \times 5.5}}$ should be input as

'5.5, c4,<tab>*; c1, c2,<tab>*;<tab>EO /, sqrt'

Example complex expression $\frac{c3}{1.1} + \frac{c0}{2.2}$ should be input as

'2.2, c0,<tab>/; 1.1, c3,<tab>/;<tab>EO +'

Example complex expression $\frac{\frac{c3+1.1}{c0+1.5}}{\frac{c5+2.2}{c9^{3.3}}}$ should be input as

'3.3, c9,<tab>^; 2.2, c5,<tab>+; 1.5, c0,<tab>+; 1.1, c3,<tab>+;<tab>EO /, /, /'

Example complex expression

$\frac{Econst}{2} * \left[\frac{c1+c3}{1-Vconst} \pm \frac{\sqrt{2}}{1+Vconst} * \sqrt{(c1-c2)^2 + (c2-c3)^2} \right]$ should be input as

'(1-Vconst)value, c1, c3,<tab>+, /; (sqrt(2)/(1+Vconst)^2value, 2, c2, c1,<tab>-, ^, *; (sqrt(2)/(1+Vconst)^2value, 2, c3, c2,<tab>-, ^, *;<tab>EO +, sqrt, +' and as

'(1-Vconst)value, c1, c3,<tab>+, /; (sqrt(2)/(1+Vconst)^2value, 2, c2, c1,<tab>-, ^, *; (sqrt(2)/(1+Vconst)^2value, 2, c3, c2,<tab>-, ^, *;<tab>EO +, sqrt, -'

15.2.5 Output Path

The output path contains the complete path to the folder where the user wants the results saved. The output path must begin and end with a '/'. This folder will

be appended if it already exists. Any existing sub-folder of the same name as a program-created folder will be overwritten. For this reason it is recommended that the folder named here be empty prior to running the program. This will allow the program to build the file structure without incident. The user must have the proper permissions to access the output directory before running the program or it will terminate in error.

The user may optionally select '*' as the output path when converting data files. This output path will write output files into the same directories as the input data files. This option is very useful when analyzing converted data after conversion because the configuration files to this software will not need to be modified very much.

Keyword: OUTPUT_PATH

15.2.6   Example HVDAMMMConfiguration.conf

Figure 22 shows an example MMM configuration file.

15.3   HVDAPSDConfiguration.conf

The HVDAPSDConfiguration.conf file contains the parameters and settings specific to the PSD frequency domain analysis. The settings in this file are constant for the entire data block. Therefore, there need be only one copy of this file in the base directory of the analysis software. HVDAPSDConfiguration.conf includes the following parameters.

15.3.1   List of Channels to Analyze

A number in this list refers to a channel in your data file. Channel '0' (zero) refers to the first saved channel. The number of channels minus one references the last saved channel. So, if there are 16 channels, the first channel is 0 and the last is 15. The numbers can be listed in any order as long as the requested channel is less than the total number of channels, greater than or equal to 0, and exists in all data files in all conditions you wish to analyze.

Keyword: CHANNELS_TO_RUN

15.3.2   List of Filters to Apply

The filters are referred to using a list of integers.

Sheet1

OUTPUT_PATH:	/server/test/

CHANNELS_TO_RUN:

0
1
2

END

FILTERS:

0

END

VIRTUAL_CHANNEL_EXPRESSIONS:
1.1, c0,   +
3.3, c9,   +; 3.3, c11,   +;   EO *
3.3, c9,   ^; 2.2, c5,   +; 1.1, c0, +; 1.1, c3, +; EO /,/,/
END

BOX_CAR_WINDOW_SIZE: 100

Figure 22: Example HVDAMMMConfiguration.conf

- 0 - No Filtering, use raw data

- 1 - Low-Pass Filter using the Low-Pass Coefficients to remove high-frequency energy

Use only one of these two possible filter states at a time by putting only one integer in the configuration file. Each filter type will produce one spreadsheet of output per file and a summary of the filter condition. End the one line filter list with 'END'.
Keyword: FILTERS

15.3.3   Output Path

The output path contains the complete path to the folder where the user wants the results saved. The output path must begin and end with a '/'. This folder will be appended if it already exists. Any existing sub-folder of the same name as a program-created folder will be overwritten. For this reason it is recommended that the folder named here be empty prior to running the program. This will allow the program to build the file structure without incident. The user must have the proper permissions to access the output directory before running the program or it will terminate in error.
Keyword: OUTPUT_PATH

15.3.4   Highest Frequency of Interest Selection

The Highest Frequency Of Interest (HFI) is used to scale the output PSD. The PSD can have frequencies beyond the HFI, but you should be certain of the samples per cycle at the HFI because of the OVER_SAMPLING input. Note: The frequencies referenced here are relative to the SAMPLE_RATE in HVDAGlobalConfiguration.conf. Choose either model or full scale units by changing the sample rate, scale factors, and these frequencies.
Keyword: HIGHEST_FREQUENCY_OF_INTEREST

15.3.5   Over Sampling Selection

The Over Sampling (OS) is used to scale the output PSD. The HFI and OS determine the re-sampling, i-skip, factor needed to give OS samples per cycle at the HFI in the output PSD.
Keyword: OVER_SAMPLING

15.3.6   Target Delta Frequency Selection

The Target Delta Frequency (TDF) is used to compute the number of points needed to calculate the PSD to arrive at the closest possible actual delta frequency to the TDF. You must determine, by hand, a good TDF for your data set. You must decide on the best trade-off between accuracy and resolution. You may want to choose a TDF that will cause a PSD data point to be calculated at a known frequency, such as the frequency when wave length is equal to ship length.

Note: If the HFI, OS, TDF combination requires a large number of data points, some data files may not have enough data. In this case, the input data are duplicated until enough data points exist to perform the desired computations. The effect of this solution is minimal on the PSD because the statistics of the input waves are not changed. The resolution you get from a duplicated data PSD is artificial. Its added resolution comes from the assumption that the data run would have continued with identical frequency content for as long as needed. This assumption affects confidence bounds statistics. This is sometimes necessary to get every single PSD from each condition to have the same frequencies, which makes summarizing and reporting much easier for the user.

Keyword: TARGET_DELTA_FREQUENCY

15.3.7   Example HVDAPSDConfiguration.conf

Figure 23 shows an example PSD configuration file.

15.4   HVDARAOConfiguration.conf

The HVDARAOConfiguration.conf file contains the parameters and settings specific to the RAO frequency domain analysis. The settings in this file are constant for the entire data block. Therefore, there need be only one copy of this file in the base directory of the analysis software. HVDARAOConfiguration.conf includes the following parameters.

15.4.1   List of Channels to Analyze

A number in this list refers to a channel in your data file. Channel '0' (zero) refers to the first saved channel. The number of channels minus one references the last saved channel. So, if there are 16 channels, the first channel is 0 and the last is 15. The numbers can be listed in any order as long as the requested channel is less

Sheet1

```
OUTPUT_PATH:                                      /server/test/

CHANNELS_TO_RUN:
                                              0
                                              1
                                              2
END

FILTERS:
                                              0
END

HIGHEST_FREQUENCY_OF_INTEREST:                 0.44
OVER_SAMPLING:                                   10
TARGET_DELTA_FREQUENCY:                           0
CONVERT_FROM_ENCOUNTER_TO_WAVE_FREQUENCY:        0
```

70

Figure 23: Example HVDAPSDConfiguration.conf

than the total number of channels, greater than or equal to 0, and exists in all data files in all conditions you wish to analyze.

Keyword: CHANNELS_TO_RUN

15.4.2 List of Filters to Apply

The filters are referred to using a list of integers.

- 0 - No Filtering, use raw data

- 1 - Low-Pass Filter using the Low-Pass Coefficients to remove high-frequency energy

Use only one of these two possible filter states at a time by putting only one integer in the configuration file. Each filter type will produce one spreadsheet of output per file and a summary of the filter condition. End the one line filter list with 'END'.

Keyword: FILTERS

15.4.3 List of RAO Filters to Apply

The filters are referred to using a list of integers.

- 0 - No Filtering, use raw RAO

- 1 - Low-Pass Filter the output RAO using the RAO Pass Coefficients to remove high-frequency oscillations in high-resolution RAOs

Use only one of these two possible filter states at a time by putting only one integer in the configuration file. Each filter type will produce one spreadsheet of output per file and a summary of the filter condition. End the one line filter list with 'END'.

Keyword: RAOFILTERS

15.4.4 List of RAO Pass Coefficients

Between the RAO filter coefficients keyword and 'END', a list of calculated filter coefficients must be copied, one coefficient per line, for each activated filter. These coefficients must be created using another program like Wave Metrics IGOR Pro IFDL and copied into the configuration file.

Keyword(s): RAO_PASS_COEFFICIENTS

15.4.5 Output Path

The output path contains the complete path to the folder where the user wants the results saved. The output path must begin and end with a '/'. This folder will be appended if it already exists. Any existing sub-folder of the same name as a program-created folder will be overwritten. For this reason it is recommended that the folder named here be empty prior to running the program. This will allow the program to build the file structure without incident. The user must have the proper permissions to access the output directory before running the program or it will terminate in error.

Keyword: OUTPUT_PATH

15.4.6 Highest Frequency of Interest Selection

The Highest Frequency of Interest (HFI) is used to scale the output RAO. The RAO can have frequencies beyond the HFI, but you should be certain of the samples per cycle at the HFI because of the OVER_SAMPLING input. Note: The frequencies referenced here are relative to the SAMPLE_RATE in HVDAGlobalConfiguration.conf. Choose either model or full scale units by changing the sample rate, scale factors, and these frequencies.

Keyword: HIGHEST_FREQUENCY_OF_INTEREST

15.4.7 Over Sampling Selection

The Over Sampling (OS) is used to scale the output RAO. The HFI and OS determine the re-sampling, i-skip, factor needed to give OS samples per cycle at the HFI in the output RAO.

Keyword: OVER_SAMPLING

15.4.8 Target Delta Frequency Selection

The Target Delta Frequency (TDF) is used to compute the number of points needed to calculate the PSD to arrive at the closest possible actual delta frequency to the TDF. You must determine, by hand, a good TDF for your data set. You must decide on the best trade-off between accuracy and resolution. You may want to choose a TDF that will cause a PSD data point to be calculated at a known frequency, such as the frequency when wave length is equal to ship length.

Note: If the HFI, OS, TDF combination requires a large number of data points, some data files may not have enough data. In this case, the input data are dupli-

cated until enough data points exist to perform the desired computations. The effect of this solution is minimal on the PSD because the statistics of the input wave are not changed. The resolution you get from a duplicated data RAO is artificial. Its added resolution comes from the assumption that the data run would have continued with identical frequency content for as long as needed. This assumption affects confidence bounds statistics. This is sometimes necessary to get every single RAO from each condition to have the same frequencies, which makes summarizing and reporting much easier for the user.

Keyword: TARGET_DELTA_FREQUENCY

15.4.9 Divisor Data Channel Selection

The Divisor Data Channel (DDC) is used to divide every PSD calculated for a given data file. Dividing a PSD by the PSD of the DDC creates a transfer function called an RAO. In ship structures, if you divide the PSD(vertical bending) / PSD(wave height) then the RAO output is in terms of bending response per foot of input wave height. The units of the transfer function RAO depend on the units of the input data and are not limited to ship structure data.

Keyword: DIVISOR_DATA_CHANNEL

15.4.10 Convert From Encounter to Wave Frequency Selection

Enter a '1' to convert from encounter frequency to wave frequency or a '0' to skip conversion. This enables the use of speed and heading from each condition's local configuration. A speed of 0.0 with a heading of 0.0 will result in the wave frequency equal to the encounter frequency.

Keyword: CONVERT_FROM_ENCOUNTER_TO_WAVE_FREQUENCY

15.4.11 Example HVDARAOConfiguration.conf

Figure 24 shows an example RAO configuration file.

15.5 HVDAWeibullConfiguration.conf

The HVDAWeibullConfiguration.conf file contains the parameters and settings specific to the Weibull distribution analysis. The settings in this file are constant for the entire data block. Therefore, there need be only one copy of this file

Sheet1

OUTPUT_PATH:	/server/test/

CHANNELS_TO_RUN:

```
0
1
2
```

END

FILTERS:

```
0
```

END

RAOFILTERS:

```
0
```

END

RAO_PASS_COEFFICIENTS:

```
4.95E-005
2.33E-006
2.35E-006
2.35E-006
2.34E-006
2.34E-006
2.35E-006
2.35E-006
2.33E-006
4.95E-005
```

END

HIGHEST_FREQUENCY_OF_INTEREST:	0.44
OVER_SAMPLING:	10
TARGET_DELTA_FREQUENCY:	0
DIVISOR_DATA_CHANNEL:	0
CONVERT_FROM_ENCOUNTER_TO_WAVE_FREQUENCY:	1

Page 1

74

Figure 24: Example HVDARAOConfiguration.conf

in the base directory of the analysis software. HVDAWeibullConfiguration.conf includes the following parameters.

15.5.1 List of Channels to Analyze

A number in this list refers to a channel in your data file. Channel '0' (zero) refers to the first saved channel. The number of channels minus one references the last saved channel. So, if there are 16 channels, the first channel is 0 and the last is 15. The numbers can be listed in any order as long as the requested channel is less than the total number of channels, greater than or equal to 0, and exists in all data files in all conditions you wish to analyze.

Keyword: CHANNELS_TO_RUN

15.5.2 List of Filters to Apply

The filters are referred to using a list of integers.

- 0 - No Filtering, use raw data

- 1 - Low-Pass Filter using the Low-Pass Coefficients to remove high-frequency energy

- 2 - High-Pass Filter using the High-Pass Coefficients to remove low-frequency energy

- 3 - Already Filtered data

Use only one of these possible filter states at a time by putting only one integer in the configuration file. Each filter type will produce one spreadsheet of output per file and a summary of the filter condition. End the one line filter list with 'END'.

Keyword: FILTERS

15.5.3 Output Path

The output path contains the complete path to the folder where the user wants the results saved. The output path must begin and end with a '/'. This folder will be appended if it already exists. Any existing sub-folder of the same name as a program-created folder will be overwritten. For this reason it is recommended that the folder named here be empty prior to running the program. This will allow the program to build the file structure without incident. The user must have the

proper permissions to access the output directory before running the program or it will terminate in error.

Keyword: OUTPUT_PATH

15.5.4 Weibull Calculation Type

There are currently four different Weibull calculation methods to choose from: 2-Parameter Linear Regression, 2-Parameter Moment Method, 3-Parameter Linear Regression, and 3-Parameter Moment Method.

Keyword: WEIBULL_TYPE

15.5.5 Weibull Amplitude Type

There are currently six types of amplitudes you may use Weibull analysis with: Maximums, Minimums, Maximums-Mean, Minimums-Mean, Ranges, Raw, Whipping, and Impacts. The Raw amplitude type allows you to provide your own set of amplitudes to HVDAS.

Keyword: AMPLITUDE_TYPE

15.5.6 Beta Lower Bound

This is the lower search bound when iteratively solving for Beta, the Weibull shape parameter, using the Moment Method calculation.

Keyword: BETA_LOWER_BOUND

15.5.7 Beta Upper Bound

This is the upper search bound when iteratively solving for Beta using the Moment Method calculation.

Keyword: BETA_UPPER_BOUND

15.5.8 Beta Step Size

This is the Beta increment size when iteratively solving for Beta using the Moment Method calculation.

Keyword: BETA_STEP_SIZE

15.5.9   Ship Lifetime

This specifies the expected, designed, or desired ship lifetime in years, which is used in maximum lifetime estimation.
   Keyword: SHIP_LIFETIME

15.5.10   Extreme F

This specifies the exceedance probability value to use when computing Extreme Value Analysis Of The Parent Weibull Distribution.
   Keyword: EXTREME_F

15.5.11   Blanket Amplitude Cutoff

This selects a Weibull threshold that will cut off detected amplitudes that are below the cutoff value for every filter state and amplitude type.
   Key Word: BLANKET_AMPLITUDE_CUTOFF

15.5.12   XNot Percent Step Size

This selects a step size used in xnot calculation as a percentage of the smallest detected amplitude that is above the cutoff. This setting is only used during 3-Parameter analysis.
   Key Word: XNOT_PERCENT_STEP_SIZE

15.5.13   Individual Amplitude Cutoff Lists

These cutoff lists are used to cutoff amplitudes below a threshold value for each amplitude type and filter state. The cutoff lists must specify a cutoff value for every channel in the data file even if it is not going to be analyzed. The list must end with END.
   Key Words: RAW_MAX_AMPLITUDE_CUTOFFS
   LOW_PASS_MAX_AMPLITUDE_CUTOFFS
   HIGH_PASS_MAX_AMPLITUDE_CUTOFFS
   RAW_MIN_AMPLITUDE_CUTOFFS
   LOW_PASS_MIN_AMPLITUDE_CUTOFFS
   HIGH_PASS_MIN_AMPLITUDE_CUTOFFS
   RAW_P2P_AMPLITUDE_CUTOFFS
   LOW_PASS_P2P_AMPLITUDE_CUTOFFS

HIGH_PASS_P2P_AMPLITUDE_CUTOFFS

15.5.14   Example HVDAWeibullConfiguration.conf

Figure 25 shows an example Weibull configuration file.

16   Using MPI

This parallelized computer software uses Message Passing Interface (MPI) to communicate between the various computers in a cluster of computers. The user must be familiar with a few simple Linux command line calls that enable and maintain the MPI communication interface. This documentation is written for the Local Area Multicomputer (LAM) implementation of MPI. The LAM-MPI communication software is what allows the Linux cluster to perform calculations in parallel.

16.1   lamhosts

'lamhosts' is a file containing the node names for all of the nodes available to the MPI software. If a node is not listed in this file, you cannot use it for calculations. The user can append nodes to this file from the user's home directory by typing "vi lamhosts" from the console or "gedit lamhosts" from an X Windows terminal. This brings up an editor. For 'vi' usage information, type "man vi" at the command line. The order of the node names in this file will determine the node's number in the 'mpirun' call. The server node MUST be listed first or the program will fail. The 'lamhosts' file can be found or created in the user's home directory.

16.2   recon

The 'recon' command checks the connections to the nodes in the 'lamhosts' file without actually bringing up the communications interface. The 'recon' command should be run before enabling the MPI interface. Type "recon -v lamhosts". This should be done from the user's home directory or change 'lamhosts' to be the full path name of the lamhosts file.

```
OUTPUT_PATH:                          /server/test/

CHANNELS_TO_RUN:
                                 0
                                 1
                                 2
END

FILTERS:
                                 0
END

WEIBULL_TYPE:                               0
AMPLITUDE_TYPE:                             0
BETA_LOWER_BOUND:                           0
BETA_UPPER_BOUND:                          10
BETA_STEP_SIZE:                             0
SHIP_LIFETIME:                              1
EXTREME_F:                               0.95
BLANKET_AMPLITUDE_CUTOFF:                   0
XNOT_PERCENT_STEP_SIZE:                     1

RAW_MAX_AMPLITUDE_CUTOFFS:
                                 0
                                 1
                                 2
END

LOW_PASS_MAX_AMPLITUDE_CUTOFFS:
                                 0
                                 1
                                 2
END

HIGH_PASS_MAX_AMPLITUDE_CUTOFFS:
                                 0
                                 1
                                 2
END

RAW_MIN_AMPLITUDE_CUTOFFS:
                                 0
                                 1
                                 2
END

LOW_PASS_MIN_AMPLITUDE_CUTOFFS:
                                 0
                                 1
                                 2
END

HIGH_PASS_MIN_AMPLITUDE_CUTOFFS:
                                 0
                                 1
                                 2
END

RAW_P2P_AMPLITUDE_CUTOFFS:
                                 0
                                 1
                                 2
END

LOW_PASS_P2P_AMPLITUDE_CUTOFFS:
                                 0
                                 1
                                 2
END

HIGH_PASS_P2P_AMPLITUDE_CUTOFFS:
                                 0
                                 1
                                 2
END
```

Figure 25: Example HVDAWeibullConfiguration.conf

16.3 lamboot

'lamboot' is the command that actually enables MPI communication between computers. The number assigned to each node can be determined by looking at the output from this command. Type "lamboot -v lamhosts" when the 'lamhosts' file is correct and the 'recon' command has completed successfully. The 'lamboot' command should be used from the user's home directory or change 'lamhosts' to be the full path name of the lamhosts file.

16.4 wipe

'wipe' stops MPI communication when the program is complete. Type "wipe -v lamhosts" at the command prompt. This should be done from the user's home directory or change 'lamhosts' to be the full path name of the lamhosts file.

16.5 lamclean

'lamclean' resets MPI communication and stops any runaway processes that are trying to use MPI. This should be done between large data runs or after the program has terminated in error. This can be done from any location assuming that the 'lamboot' command has already been used and MPI communication is still active. Type "lamclean -v" at the command prompt.

16.6 mpirun

'mpirun' is the command that starts a program that uses MPI in parallel. Precede the program name with "mpirun -v" followed by the first and last nodes in the 'lamhosts' file (for 4 nodes use n0-3, this uses nodes 0,1,2,3 in the 'lamhosts' file). The 'mpirun' command can use any node listed in the 'lamhosts' file. It is not required to use all the nodes listed in 'lamhosts'. A sample call to MPI using 5 nodes would be "mpirun -v n0-4 program program_args". The 'mpirun' command should be used in the folder containing the program or configuration files to be run. Another sample call to MPI using all available nodes would be "mpirun -v C program program_args"; the "C" flag tells mpirun to use all available processors.

17 Using HVDAS

17.1 General Setup

1. Make sure all of your data are backed up before trying to setup anything!

2. Make sure all of your data are backed up before trying to analyze anything!

3. Triple check your backups before moving on! You have been warned.

4. Use Wave Metrics IGOR Pro Filter Design Laboratory, or similar software package, to design appropriate finite impulse response (FIR) filters for your specific data. You may create up to 3 different FIR filters. You may create a smoothing-pass filter to remove noise, a low-pass filter to remove high-frequency energy, and a high-pass filter to remove low-frequency energy. You may also choose to box car average your data to remove drop-outs. Look at your data in a package like IGOR to make sure that your settings make sense with your current data set and desired results.

5. Save the FIR filter coefficients that describe your filters in ASCII text files. These coefficients will be pasted into the proper configuration files later.

6. Locate the file 'HVDASuite-YYYY-MM-DD.tar.gz' or 'HVDASuite.tar.gz'. An '_i386' in the file name indicates that this binary executable program was compiled and built for Intel i386 based computer architecture. Copy this file to a globally accessible working directory like '/Server/Analysis/'. Change the working directory to be that directory. Type, "tar xvzf HVDA*.tar.gz", to unpack the archive file. This will create a sub-folder, called 'HVDA-Suite', which will contain the executable program 'hvdas' and documentation.

7. Type "./" to reference the current directory in most Unix operating systems. You will need to add the './' before 'hvdas' if you only unpacked the archive and did not install the executable code into a system-wide accessible location.

8. Type "hvdas –help" or "hvdas -help" for usage information.

9. Make sure the data on the analysis computer are stored in a globally accessible location if you are going to run the analysis on a cluster. It is generally a good idea to put all of the data conditions in a small number of well-named

locations. See section 'One Valid Input Structure' for a complex data layout. The best approach is to keep it simple. Each separate condition must be in a separate sub-folder with a unique name. The simplest layout is to have one main data directory which contains one sub-folder for each of your separate conditions.

10. Create the example configuration files 'HVDAGlobalConfiguration.conf', 'HVDAMMMConfiguration.conf', 'HVDAPSDConfiguration.conf', 'HVDARAOConfiguration.conf', and 'HVDAWeibullConfiguration.conf' by typing "hvdas -createexampleconfigs".

11. Modify the 'HVDAGlobalConfiguration.conf', 'HVDAMMMConfiguration.conf', 'HVDAPSDConfiguration.conf', 'HVDARAOConfiguration.conf', and 'HVDAWeibullConfiguration.conf' configuration files. The example configuration files are well documented.

12. Add an entry to 'HVDAGlobalConfiguration.conf' in TEST_CONDITION_LIST for test condition folder. You will need to change each entry to reflect the data contained in each condition directory. Make sure you change CONDITION_LABEL. Each condition label MUST be unique or the results will be invalid. A well-designed template will not need to be modified much from condition to condition if the data is similar.

13. Be sure to paste all of the FIR filter coefficients and initial condition zero values into the 'HVDAGlobalConfiguration.conf' file. Also, be sure to get ALL of the data and header formatting information, channel names, and scale factors correct or the results will be wrong.

17.2 Command Line Options

HVDAS has several command line options. These options are very useful for scripting large sets of data to be run over an extended period of time. The options and flags can be input in any order.

17.2.1 –help or -help

'hvdas –help' or 'hvdas -help' displays usage information and command line options to the standard output.

17.2.2 -convertdata

'hvdas -convertdata' specifies that the software should convert and modify data to ASCII. This will output converted or virtual channels to new ASCII files. Old files will be overwritten without additional warning.

17.2.3 -createexampleconfigs

'hvdas -createexampleconfigs' specifies that the software should create the example configuration files that are necessary to run the desired analysis program. It will create HVDAGlobalConfiguration.conf, HVDAMMMConfiguration.conf, HVDAPSDConfiguration.conf, HVDARAOConfiguration.conf, and HVDAWeibull-Configuration.conf. Old configuration files will be overwritten if you type 'yes' after the warning.

17.2.4 -mpi

'hvdas -mpi' specifies that the software should run in MPI mode. This tells the HVDAS program to run analysis in parallel. The most basic way to start HVDAS on a cluster is 'mpirun -v C hvdas -mpi <options>' to run this program on all available nodes; or 'mpirun -v -np 2 n0 hvdas -mpi <options>' to run the client and server portions of HVDAS locally, on one singular node with one or more processors.

17.2.5 -uni

'hvdas -uni' specifies that the software should run in uniprocessor mode. This tells the HVDAS program to run analysis locally using only one processor. The most basic way to start HVDAS on a single processor computer is 'hvdas -uni <options>'.

17.2.6 -mmm

'hvdas -mmm' specifies that hvdas should perform MMM analysis. This option must be used in conjunction with either -mpi or -uni.

17.2.7   -psd

'hvdas -psd' specifies that hvdas should perform PSD analysis. This option must be used in conjunction with either -mpi or -uni.

17.2.8   -rao

'hvdas -rao specifies that hvdas should perform RAO analysis. This option must be used in conjunction with either -mpi or -uni.

17.2.9   -weibull

'hvdas -weibull specifies that hvdas should perform Weibull analysis. This option must be used in conjunction with either -mpi or -uni.

17.2.10   -version

'hvdas -version' will display version information.

17.2.11   -debugoutput

'-debugoutput' is a flag that tells HVDAS to output extra textual information to the standard output, for debugging purposes. The default for this option is false.

17.2.12   -hvdaglobalconfigurationname

'-hvdaglobalconfigurationname' lets you specify the full path name of the global configuration file. The default is "./HVDAGlobalConfiguration.conf".

17.2.13   -hvdammmconfigurationname

'-hvdammmconfigurationname' lets you specify the full path name of the MMM configuration file. The default is "./HVDAMMMConfiguration.conf".

17.2.14   -hvdapsdconfigurationname

'-hvdapsdconfigurationname' lets you specify the full path name of the PSD configuration file. The default is "./HVDAPSDConfiguration.conf".

17.2.15   -hvdaraoconfigurationname

'-hvdaraoconfigurationname' lets you specify the full path name of the RAO configuration file. The default is "./HVDARAOConfiguration.conf".

17.2.16   -hvdaweibullconfigurationname

'-hvdaweibullconfigurationname' lets you specify the full path name of the Weibull configuration file. The default is "./HVDAWeibullConfiguration.conf".

17.2.17   -logfilename

'-logfilename' lets you specify the full path name of the log file you would like to output. The default is "./HVDASLog.log".

17.2.18   -nofilestats

'-nofilestats' tells HVDAS that it should not output results files for every single input file. This flag will restrict output to condition summary results only. This option is useful if you have a lot of data files in each condition and you do not care about the individual data files. The default is false, which means that HVDAS will output individual file and condition results.

17.2.19   -nosummarystats

'-nosummarystats' tells HVDAS that it should not output results files for the condition summary. This flag will restrict output to individual file results only. This option is useful if your condition directories have only one data file. The default is false, which means that HVDAS will output individual file and condition results.

17.2.20   -weibullfilterstate

'-weibullfilterstate' lets you specify the filter to use for Weibull analysis and will override the setting in the HVDAWeibullConfiguration.conf configuration file.

17.2.21   -weibullamplitudetype

'-weibullamplitudetype' lets you specify the set of amplitudes to use for Weibull analysis and will override the setting in the HVDAWeibullConfiguration.conf configuration file.

17.3   Cluster Execution

1. Type: "hvdas –help" for usage information.

2. Warning: Old results in the results output path will be overwritten without warning!

3. Triple check your results output path to make sure you are not going to overwrite anything important! You have been warned.

4. Go to the home directory "cd ~".

5. Edit 'lamhosts' if necessary "vi lamhosts".

6. Recon 'lamhosts' Type "recon -v lamhosts" to see if the cluster is boot-able.

7. If 'recon' is successful, bring up MPI using "lamboot -v lamhosts".

8. If 'lamboot' is successful, go to the analysis directory where the HVDAS configuration files are located. For example: Type "cd /server/test/".

9. If all the configuration files are set up and 'lamboot' worked, run the program: "mpirun -v C hvdas -mpi <options>", which will actually perform the desired analysis. Adjust the "C" to "n0-5" to use a smaller subset of available nodes. In one example case, 6 computers are in the 'lamhosts' file, 0 is the server and 1 through 5 are the compute nodes, so specifying "C" is exactly the same as specifying "n0-5". You may optionally run 'hvdas -convertdata' to smooth, filter, scale, zero, or reformat your input data files and save them in new ASCII tab-delimited text data files.

10. To perform MMM analysis run "mpirun -v C hvdas -mpi -mmm -logfilename mmmlog.log".

11. To perform PSD analysis run "mpirun -v C hvdas -mpi -psd -logfilename psdlog.log".

12. To perform RAO analysis run "mpirun -v C hvdas -mpi -rao -logfilename raolog.log".

13. To perform Weibull analysis run "mpirun -v C hvdas -mpi -weibull -logfilename weibulllog.log".

14. If errors are reported, fix and rerun.

15. When program has completed, clean MPI "lamclean -v".

16. The MMM analysis results should be in the '/server/test/HVDAMMMResults/' folder in the location chosen in the 'HVDAMMMConfiguration.conf' file. The converted or modified data files should be in the '/server/test/HVDAConvertedOutputData/' folder in the same location chosen in the 'HVDAMMMConfiguration.conf' file.

17. If another analysis is desired, change the necessary settings and rerun, otherwise shutdown MPI from the home directory using "wipe -v lamhosts".

18. If everything worked properly, you should see messages telling you that everything finished properly. You should see the elapsed time and the location of the results.

19. A log file called 'HVDASLog.log' should have been generated unless the -logfilename option was used. This log file contains debugging information, analysis information, progress information, and error information. It is a good idea to check this file to make sure everything went as planned and to check for errors.

20. Check the results output folder. The 'HVDA*Results/' folder should contain one sub-folder for each input condition. Check these folders for individual output files and condition summaries. The output is standard ASCII tab delimited text, which can be imported into almost any spreadsheet and by any ASCII text editor.

21. It is a good idea to manually check the results for three or more channels, two or more filter states, in three or more conditions to ensure good output.

22. If you got good results, back them up!

23. Back up your results! You have been warned.

17.4   Workstation (Single Processor) Execution

1. Type: "hvdas –help" for usage information.

2. Warning: Old results in the results output path will be overwritten without warning!

3. Triple check your results output path to make sure you are not going to overwrite anything important! You have been warned.

4. Go to the analysis directory where the HVDAS configuration files are located. For example: Type "cd /server/test/".

5. To perform MMM analysis run "hvdas -uni -mmm -logfilename mmm-log.log".

6. To perform PSD analysis run "hvdas -uni -psd -logfilename psdlog.log".

7. To perform RAO analysis run "hvdas -uni -rao -logfilename raolog.log".

8. To perform Weibull analysis run "hvdas -uni -weibull -logfilename weibullog.log".

9. To perform data conversion run "hvdas -convertdata -logfilename convert-log.log".

10. If errors are reported, fix and rerun.

11. The MMM analysis results should be in the '/server/test/HVDAMMMResults/' folder in the location chosen in the 'HVDAMMMConfiguration.conf' file. The converted or modified data files should be in the '/server/test/HVDAConvertedOutputData/' folder in the same location chosen in the 'HVDAMMMConfiguration.conf' file.

12. If another analysis is desired, change the necessary settings and rerun.

13. If everything worked properly, you should see messages telling you that everything finished properly. You should see the elapsed time and the location of the results.

14. A log file called 'HVDASLog.log' should have been generated unless the -logfilename option was used. This log file contains debugging information, analysis information, progress information, and error information. It is a good idea to check this file to make sure everything went as planned and to check for errors.

15. Check the results output folder. The 'HVDA*Results/' folder should contain one sub-folder for each input condition. Check these folders for individual output files and condition summaries. The output is standard ASCII

tab delimited text, which can be imported into almost any spreadsheet and by any ASCII text editor.

16. It is a good idea to manually check the results for three or more channels, two or more filter states, in three or more conditions to ensure good output.

17. If you got good results, back them up!

18. Back up your results! You have been warned.

17.5   Viewing Results

17.5.1   Importing ASCII Text Files Into a Spreadsheet

A great way to look at the result files from HVDAS is with a spreadsheet program such as Microsoft Excel, Sun Microsystems Star Office, or Open Office. Try to open the result file as an ASCII text file or Comma Separated Value (CSV) text file. Most spreadsheet programs will allow you to select some options to describe the text file. Make sure that you select a Tab character as the separating or delimiting character if it is not the default.

17.5.2   Printing ASCII Text Files From Linux/UNIX

1. Go to the directory containing the data to be printed.

2. Type: "a2ps –landscape –columns 1 –rows 1 -f 8 -d Output.txt" or open the results in a spreadsheet program and print from there.

- Option '–landscape' Sets the orientation of the printed page

- Option '–columns' Sets the number of pages to print vertically per-page

- Option '–rows' Sets the number of pages to print horizontally per-page

- Option '-f' Sets the font size, 8 in this example

- Option '-d' Precedes the file name to print

- The file name can include *'s to print multiple files at once.

- The output results files are tab delimited and may be imported into various spreadsheets easily.

17.5.3 Using gnuplot to View Data

1. Go to the directory containing the data to be printed.

2. Type: "gnuplot" to start the gnuplot command interpreter or type: "gnuplot scriptfile.gnuplot" to have gnuplot read commands from a text file.

3. gnuplot can read ASCII text data files if all of the header lines begin with "#" comment characters

4. Command: "plot "Data_mod_0.txt" using 1 title 'ASDSD19S38ch 1' with lines, "smooth1.txt" using 1 with lines lw 3" will plot channel 1 from Data_mod_0.txt using lines, and channel 1 from smooth1.txt using lines with line width of 3.

5. Command: "set out "pstest.ps"" will set the output file to "pstest.ps"

6. Command: "set size 1.0, 0.5" will set the size of the output plot

7. Command: "set terminal postscript landscape enhanced "Helvetica" 12" will set the type of output, font, and font size

8. Steps to print out plots on paper:

- set size 1.0, 0.5

- set terminal postscript landscape enhanced "Helvetica" 12

- !rm -f pstest.ps

- set out "pstest.ps"

- plot "Data_mod_0.txt" using 1 title 'ASDSD19S38ch 1' with lines, "smooth1.txt" using 1 with lines lw 3

- !lpr pstest.ps

References

[1] Hardin, Jay C. "Introduction to Time Series Analysis." NASA Reference Publication 1145. NASA Publishing, 1986.

[2] Press, William H., Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. "The Art Of Scientific Computing." Numerical Recipes In C Second Edition. New York: Cambridge University Press, 1992.

[3] IGOR Pro Team. "Technical Note 006." Wave Metrics IGOR Pro 3.15 Manual. 2000.

[4] Weast, Robert C., Samuel M. Selby. CRC Handbook of Math 4TH Edition. Cleveland: The Chemical Rubber Co., 1970.

[5] Abramowitz, Milton, Irene A. Stegun. Handbook of Mathematical Functions. New York: Dover Publications, 1972.

[6] Lewis, Richard R. "An Assessment Of Statistical Techniques And Current Assumptions For Analysis Of Hull Girder Bending Response Time Histories." Survivability, Structures and Materials Directorate Research and Development Report. Bethesda: Naval Surface Warfare Center Carderock Division, 1995.

[7] Schwartz, David A., Randall R. Judd, William J. Harrod, Dwight P. Manley. VSIPL Spec 1.0. Georgia: Georgia Tech Research Corporation, 2000.

[8] Message Passing Interface Forum. MPI-2: Extensions to the Message-Passing Interface. Knoxville: University of Tennessee, 1997.

[9] Ohio Supercomputer Center. MPI Primer / Developing With LAM. Ohio, Ohio State University, 1996.

[10] Radajewski, Jacek, Douglas Eadline. Beowulf HOWTO Version 1.1.1. On-line PDF File. 1998.

[11] Radajewski, Jacek, Douglas Eadline. Beowulf Installation And Administration HOWTO Version 0.1.2. On-line PDF File. 1999.

[12] Bendat, Julius C., Allan G. Piersol. Random Data: Analysis And Measurement Procedures. New York: Wiley-Interscience, 1971.

[13] Society Of Naval Architects And Marine Engineers. Principles of Naval Architecture. New York: The Society Of Naval Architects And Marine Engineers, 1967.

[14] Newland D. E. An Introduction To Random Vibrations and Spectral Analysis Second Edition. New York: Longman Inc., 1984.

[15] Sikora, Jerome P., Alfred Dinsenbacher, Jeffrey E. Beach. "A Method For Estimating Lifetime Loads And Fatigue Lives For Swath And Conventional Monohull Ships." Naval Engineers Journal. May 1983.

[16] Abernethy, Dr. R. B., J. E. Breneman, C. H. Medlin, G. L. Reinman. Weibull Analysis Handbook. Wright-Patterson Air Force Base, Ohio: Air Force Systems Command, 1983.

[17] Sikora, Jerome P. "Cumulative Lifetime Load Criteria For Surface Ships." Ship Structures and Protection Department Departmental Report. Bethesda: Naval Surface Warfare Center Carderock Division, 1989.

[18] Dalzell, John F. "The Ship in a Seaway." NSWCCD Technical Report. Bethesda: Naval Surface Warfare Center Carderock Division, 1970.